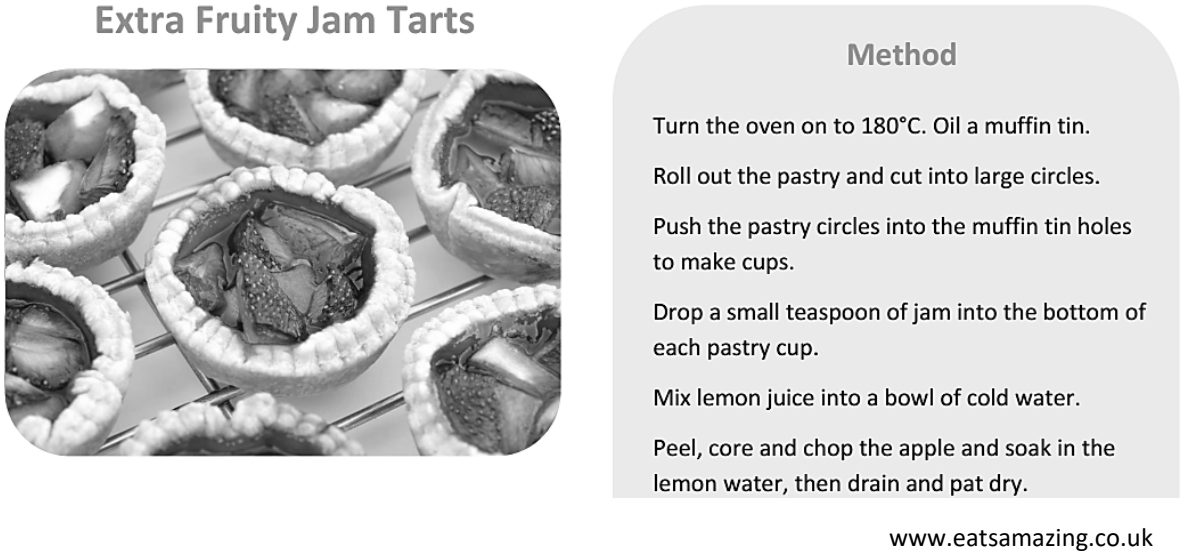
13 Algorithmau ar gyfer datrys problemau

Mae diffiniad ehangach o rifedd yn ymestyn ymhell y tu hwnt rhifyddeg a chymhwyso rhif. Mae sgiliau rhifedd sy'n cael eu gwerthfawrogi'n fawr gan gyflogwyr yn cynnwys: datrys problemau mewn cyd-destun galwedigaethol, y gallu i weithio gyda systemau technoleg gwybodaeth, a'r gallu i drosi rhwng gwahanol gynrychioliadau o ddata meintiol gan ddefnyddio rhifau, diagramau neu fynegiadau algebraidd fel y bo'n briodol. Mae pob un o'r sgiliau ehangach hyn yn dod ynghyd yn y dyluniad o **algorithmau**.

Mae algorithm yn set o gyfarwyddiadau ar gyfer gwneud tasg. Gallai hyn fod yn rhywbeth mor gyfarwydd ag rysáit coginio:



Ffigur 415: Algorithm ar gyfer gwneud tartenni jam

Fodd bynnag, rydym yn tueddu i ddefnyddio'r term algorithm amlaf wrth gyfeirio at ddilyniannau o gyfarwyddiadau mewn rhaglenni cyfrifiadurol. Yn nodweddiadol, mae algorithm yn cynnwys cyfrifiad amlgam, a gall ei ddyluniad yn gofyn am lefel uchel o rifedd a sgiliau datrys problemau.

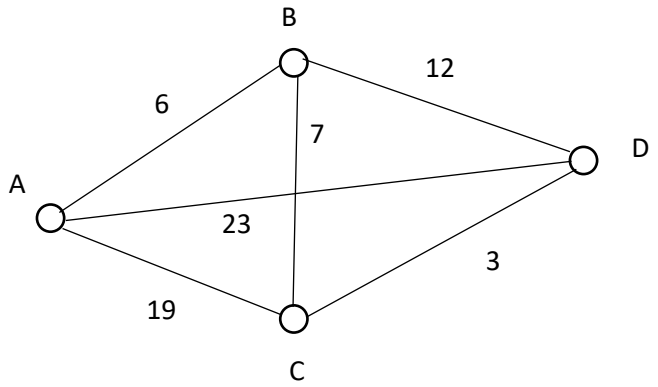
Wrth gyflwyno'r cynllun ar gyfer algorithm, efallai y bydd y camau yn cael eu hysgrifennu mewn iaith gyffredin, fel yn yr enghraifft o rysáit uchod, neu gellir ei harddangos mewn ffurf siart llif neu fath arall o ddiagram.

Mae'r defnydd o algorithmau mathemategol mewn rhaglenni cyfrifiadurol wedi dod yn fwy ac yn fwy pwysig wrth i systemau cyfrifiadurol wedi dod yn fwy pwerus. Gall meddalwedd yn awr yn cyflawni llawer o'r tasgau cymhleth a oedd yn bosibl dim ond drwy gudd-wybodaeth ddynol o'r blaen. Enghreifftiau y byddwn yn ystyried yn y bennod hon yn cynnwys: cynllunio teithiau, didoli data yn gyflym, chwarae gemau, ac amgryptio data.

Rydym yn dechrau drwy ymchwilio algorithmau a ddefnyddir mewn cynllunio taith.

Algorithm Dijkstra

Gofyniad aml mewn cymwysiadau cyfrifiadurol yw dod o hyd i lwybr byrraf, gyflymaf, neu rataf o un lle i'r llall drwy rwydwaith o lwybrau posibl. Er mwyn dangos y dasg hon, byddwn yn ystyried rhwydwaith bach:

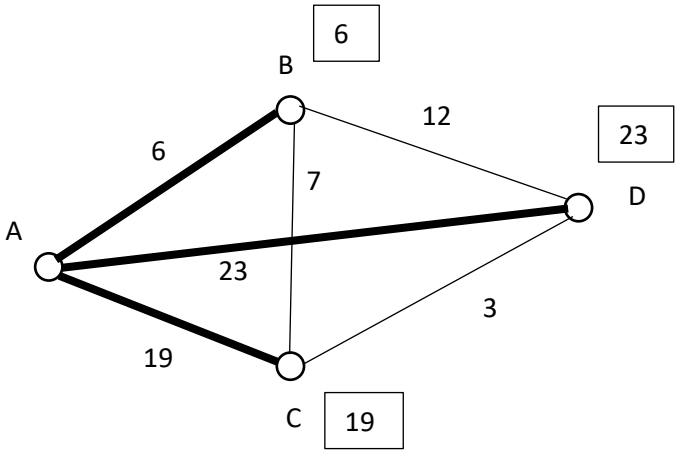


Ffigur 416: Rwydwaith o drefi a phellteroedd teithio

Mae'r **nodau** yn y rhwydwaith, A - D yn cynrychioli pedair tref. Mae'r **cysylltiadau** rhwng y nodau yn cynrychioli ffyrdd, gyda phellteroedd a ddangosir yn km. Mae'r rhwydwaith yn cael ei gynrychioli ar ffurf diagram, felly nid yw'r cysylltiadau yn cael eu tynnu i raddfa. Mae gennym ddiddordeb yn syml yn dopoleg y rhwydwaith: y ffordd y mae'r nodau yn cael eu cysylltu.

Amcan y dasg yw dod o hyd i'r llwybr byrraf o dref A i dref D, allan o'r gwahanol lwybrau sydd yn bosibl. Pe baem yn gwneud hyn â llaw, efallai y byddwn yn gwneud ychydig o gyfrifiadau syml trwy rifyddeg pen cyn penderfynu ar yr ateb. Fodd bynnag, petai'r dasg yw cael ei wneud gan gyfrifiadur, yna mae'n rhaid nodi algorithm penodol iawn ar gyfer gwirio pob llwybr posibl. Ddull effeithlon am gyflawni'r dasg hon yw **algorithm Dijkstra**. Gall yr un algorithm yn cael ei ddefnyddio gan y cyfrifiadur i ddod o hyd i atebion yn gyflym i broblemau canfod llwybr sydd yn llawer mwy cymhleth a heriol, megis dewis y llwybr byrraf i deithio ar hyd ffyrdd o Gaerdydd, drwy Dwnnel y Sianel, i Budapest!

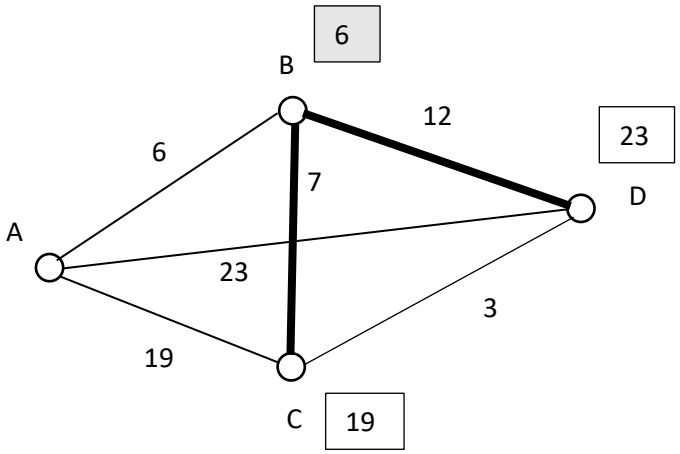
Wrth ddychwelyd at ein rhwydwaith syml, y cam cyntaf yw adnabod y trefi y gellir eu cyrraedd yn uniongyrchol o'r man cychwyn. Yna byddwn yn cofnodi'r pellter teithio ar hyd y dolenni.



Ffigur 417: Cysylltiadau o dref A

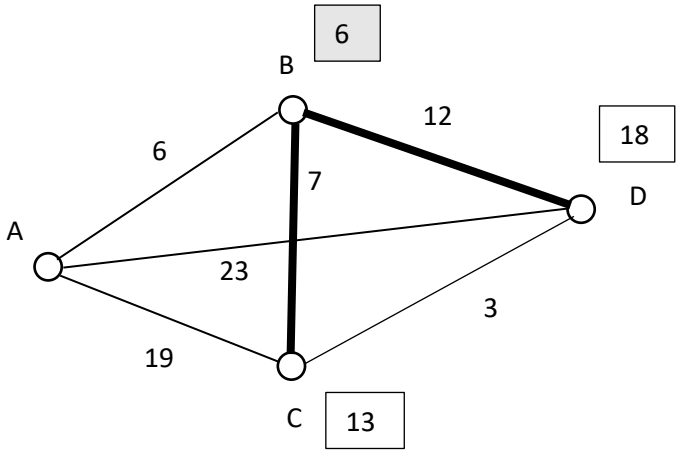
Sylwch fod y pellteroedd yn atebion dim ond **dros dro** ar hyn o bryd. Gall fod yn bosibl i gyrraedd rhai o'r trefi gan lwybr byrrach. (Mae hyn yn wir am dref C, y gellid ei gyrraedd mewn dim ond 13 km wrth deithio trwy dref B.)

Rydym bellach yn edrych ar y pellteroedd dros dro ac yn dewis yr isaf o'r rhain. Mae hyn yn bellter o 6 km i dref B. Mae'n rhaid i hwn fod y pellter byrraf i'r dref B. Byddai unrhyw lwybr arall i dref B yn golygu teithio trwy un o'r nodau eraill ac mae eu pellter nhw eisoes yn fwy na 6 km o'r man cychwyn. Felly gallwn wneud y pellter i'r dref B yn ateb parhaol. Rydym bellach yn edrych ar lwybrau o'r dref B i unrhyw nodau eraill sy'n dal yn ymddangos pellteroedd dros dro:



Ffigur 418: Cysylltiadau o dref B

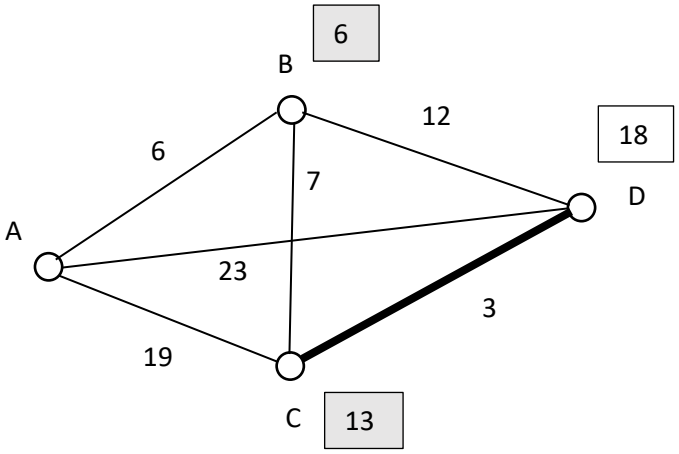
Os byddwn yn teithio i dref D drwy dref B, byddai cyfanswm y pellter fod yn 18 km. Mae hyn yn welliant ar werth blaenorol o 23 km, fel y gallwn ddiweddaru nôd D. Yn yr un modd, mae'n ddim ond 13 km i gyrraedd tref C trwy dref B, sy'n welliant ar werth blaenorol o 19 km. Mae'r diagram nawr yn dyfod:



Ffigur 419: Pellteroedd diweddaru drwy dref B

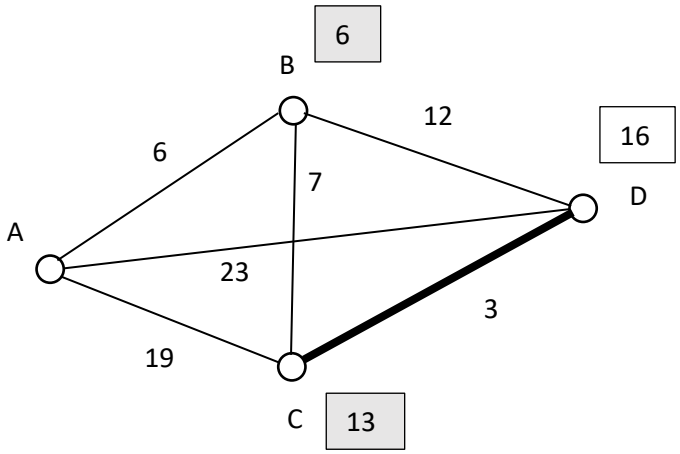
Gall yr algorithm yn awr yn cael ei ailadrodd. Rydym yn chwilio am y lleiaf o'r pellteroedd dros dro, sydd nawr 13 km i dref C. Mae'n ***rhaid*** i hwn fod y pellter byrraf o'r man cychwyn

i'r dref C. Yr unig opsiwn arall nad ydym wedi ystyried eto yw cyrraedd tref C trwy dref D. Ni allai hyn roi llwybr byrrach, gan fod y pellter i D eisoes yn fwy. Gallwn felly gofnodi'r pellter byrraf i'r dref C fel **parhaol**. Yna byddwn yn chwilio am gysylltiadau o dref C i unrhyw nodau eraill sy'n dal i gael pellter dros dro eu nodi.



Ffigur 420: Cysyllt o ddref C

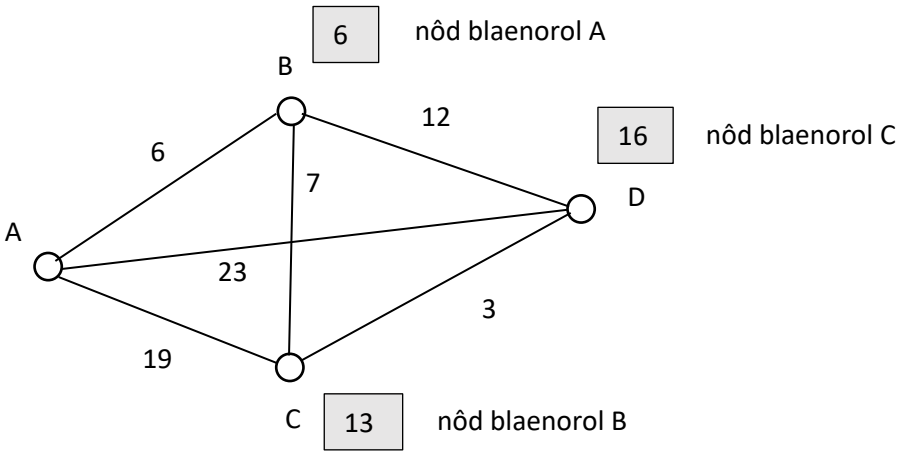
Dim ond y cyswllt i'r dref D yn aros i gael eu prosesu. Rydym yn cyfrifo y gallai nôd hwn ei gyrraedd mewn 16 km trwy dref C, sy'n welliant ar y pellter blaenorol o 18 km. Gall y nôd, felly, yn cael ei diweddaru.



Ffigur 421: Pellter wedi diweddaru i dref D drwy dref C

Tref D yw'r isaf (ac yr unig) nôd dros dro sydd ar ôl, fel y gall y pellter o 16 km yn cael ei wneud yn barhaol.

Rydym nawr wedi cyfrifo yn gywir y pellter byrraf posibl o'r man cychwyn i bob un o'r nodau eraill yn y rhwydwaith, ond sut y gall hyn helpu gyda'n tasg wreiddiol i ddod o hyd i'r llwybr byrraf o dref A i dref D? Yr ateb yw cadw cofnod o'r nôd blaenorol wrth i bob pellter yn cael ei gofnodi neu ei ddiweddaru:



Ffigur 422: Nodau blaenorol ar gyfer pellteroedd teithio byrraf i bob tref

Gan weithio yn ôl oddi wrth y gyrchfan D, gwelwn y cafodd hyn ei gyrraedd o dref C. Cyrhaeddwyd tref C o dref B, a daethpwyd tref B o'r man cychwyn A. Felly, mae'r llwybr byrraf wedi ei ganfod:

dref A → dref B → dref C → dref D

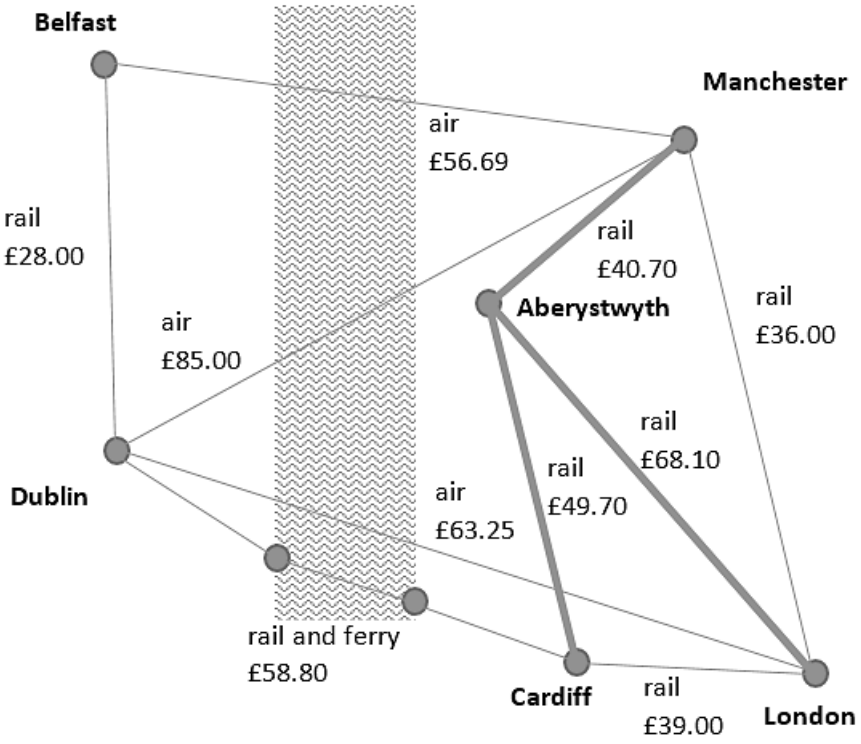
Gadewch i ni nawr yn ystyried enghraifft fwy realistig:

> Mae myfyriwr yn dymuno teithio o Aberystwyth i Ddulyn ym modd rhataf. Opsiynau posibl yw:
>
> - Teithio ar y trên i faes awyr Manceinion. Hedfan yn uniongyrchol i Ddulyn, neu hedfan i Belfast a chwblhau'r daith ar y trên.
> - Teithio ar y trên i faes awyr Llundain, ac yna hedfan i Ddulyn.
> - Teithio ar y trên i Gaerdydd, yna cwblhau'r daith i Ddulyn ar y trên a fferi.

Prisiau tocyn ar gyfer gwahanol rannau o'r llwybrau posibl yn cael eu dangos yn y ffigur 423 isod.

Byddai cyfrifiadur yn defnyddio tabl data wrth ddatrys y broblem. Rydym yn dechrau drwy restru'r nodau'r rhwydwaith. Mae colofnau ychwanegol yn cael eu darparu yn y tabl ar gyfer **pris teithio** i bob nôd, **statws** fel dros dro, neu barhaol os gwyddom nawr fod hyn y pris isaf posibl. Mae colofn hefyd i gofnodi'r nôd **blaenorol** ar hyd y llwybr.

NÔD	PRIS TEITHIO	STATWS	BLAENOROL
Aberystwyth	0	parhaol	
Belfast	-	dros dro	
Caerdydd	-	dros dro	
Dulyn	-	dros dro	
Llundain	-	dros dro	
Manceinion	-	dros dro	



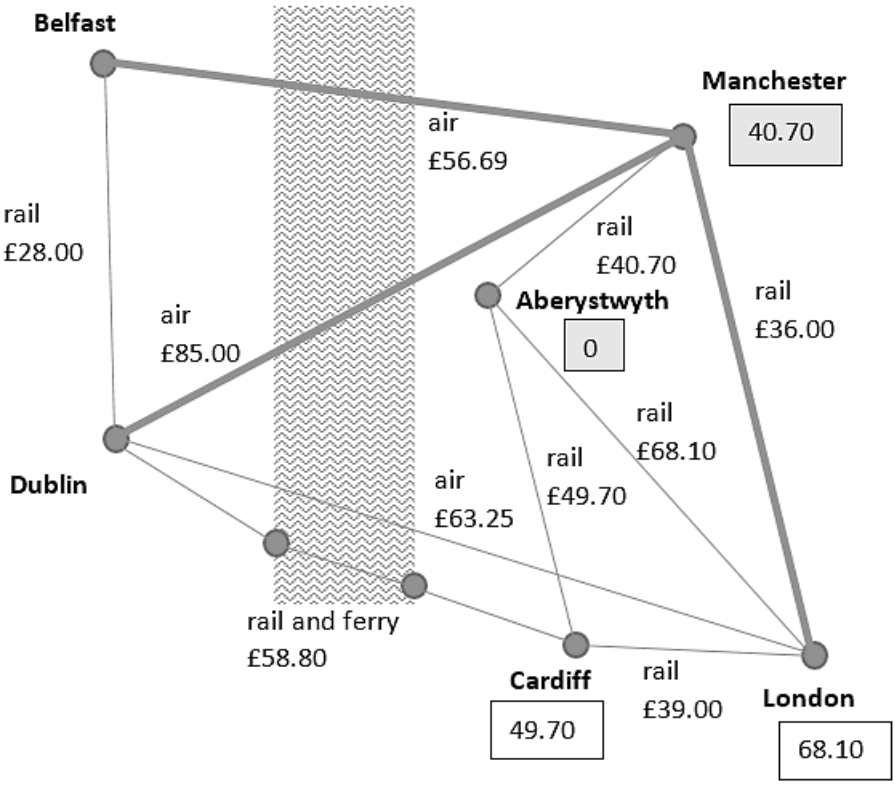
Ffigur 423: Opsiynau pris ar gyfer teithio o Aberystwyth i Ddulyn

Rydym yn dechrau yn ***Aberystwyth***. Mae cysylltiadau i ***Gaerdydd***, ***Llundain*** a ***Manceinion***. Rydym yn cofnodi'r prisiau mewn pob achos, ac yn gosod y ***nôd blaenorol*** ar gyfer pob un o'r lleoliadau hyn fel ***Aberystwyth***.

NÔD	PRIS TEITHIO	STATWS	BLAENOROL
Aberystwyth*	0	parhaol	
Belfast	-	dros dro	
Caerdydd	49.70	dros dro	Aberystwyth
Dulyn	-	dros dro	
Llundain	68.10	dros dro	Aberystwyth
Manceinion	40.70	dros dro	Aberystwyth

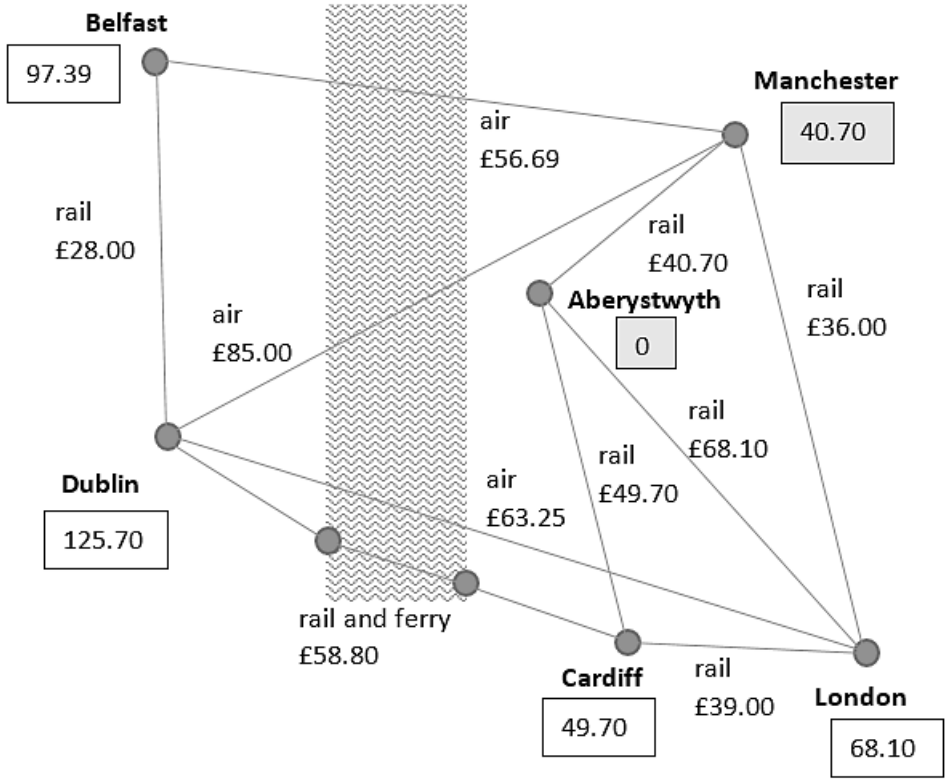
Rydym yn canfod ***Manceinion*** fel y nôd dros dro gyda'r gost isaf o £40.70. Mae hyn yn cael ei osod i ***barhaol***, ac yn dod yn ***nôd cyfredol***.

NÔD	PRIS TEITHIO	STATWS	BLAENOROL
Aberystwyth	0	parhaol	
Belfast	-	dros dro	
Caerdydd	49.70	dros dro	Aberystwyth
Dulyn	-	dros dro	
Llundain	68.10	dros dro	Aberystwyth
Manceinion*	40.70	parhaol	Aberystwyth



Ffigur 424: Cysylltiadau o Fanceinion i nodau dros dro eraill

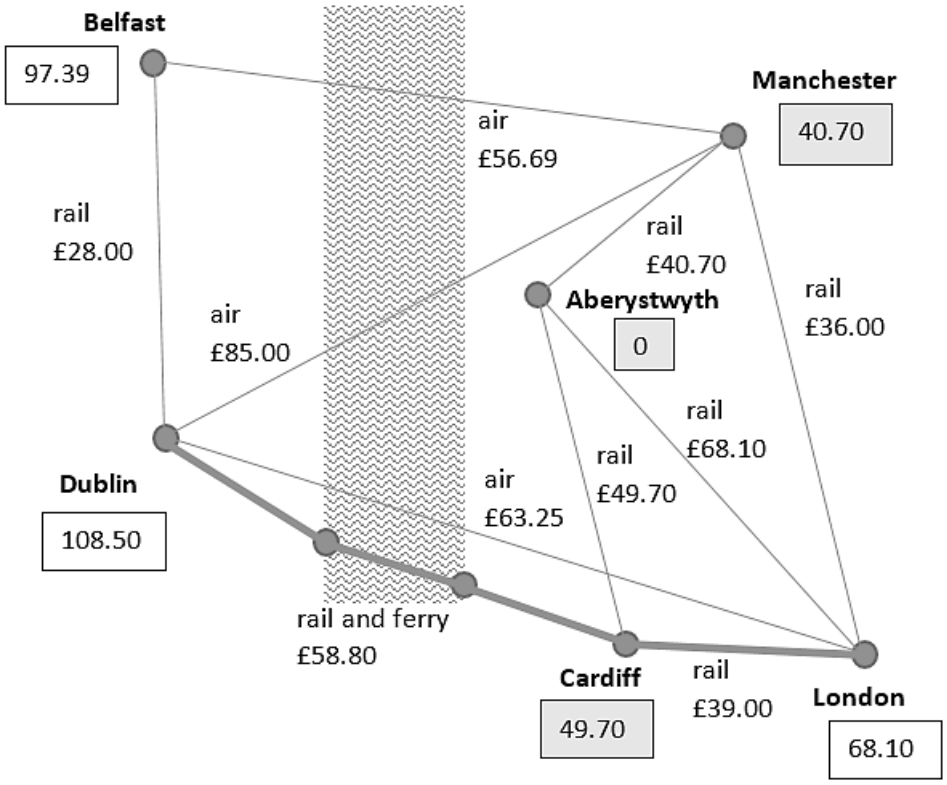
Mae cysylltiadau o Fanceinion i'r nodau dros dro ***Belfast***, ***Llundain*** a ***Dulyn***. Gallwn gyfrifo cyfanswm y costau teithio i'r trefi hyn trwy Fanceinion. Gall y costau teithio i'r Belfast a Dulyn yn cael ei gofnodi, a nodau blaenorol a osodir i Fanceinion. Fodd bynnag, byddai'r pris i Lundain drwy Fanceinion o £ 76.70 fod yn ddrutach na theithio yn uniongyrchol o Aberystwyth.



Ffigur 425: Prisiau wedi diweddaru o Fanceinion

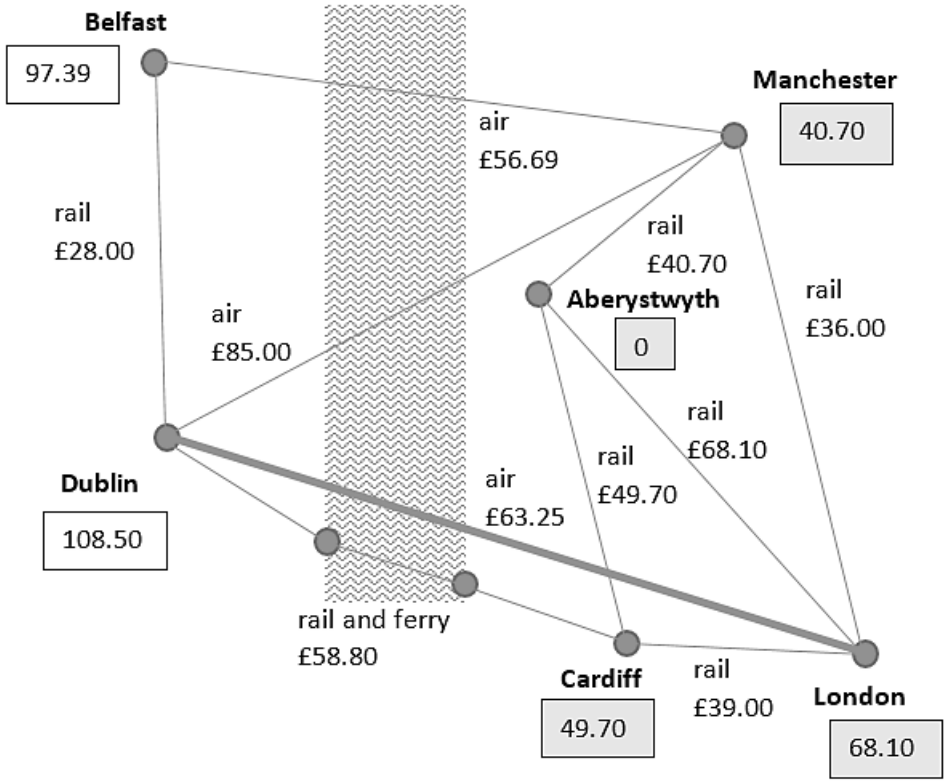
Nawr fod nôd Manceinion wedi'i brosesu, mae ***Caerdydd*** yn dod y nôd dros dro gyda'r gost isaf o £49.70. Mae hyn bellach wedi ei osod yn ***barhaol***, ac yn dod y nôd cyfredol. Mae gan Gaerdydd gysylltiadau i'r nodau dros dro ***Llundain*** a ***Dulyn***. Rydym yn cyfrifo cyfanswm y costau teithio i'r trefi hyn. Byddai'r pris i Lundain drwy Gaerdydd yn ddrutach na'r gwerth a ddangosir eisoes, felly mae hyn yn cael ei anwybyddu. Mae'r pris i Ddulyn yn llai na gwerth presennol o £125.70, felly mae'r gost yn cael ei ddiweddaru a'r ***nôd blaenorol*** yn cael ei hail-osod i Gaerdydd.

NÔD	PRIS TEITHIO	STATWS	BLAENOROL
Aberystwyth	0	parhaol	
Belfast	97.39	dros dro	Manceinion
Caerdydd*	49.70	parhaol	Aberystwyth
Dulyn	108.50	dros dro	Caerdydd
Llundain	68.10	dros dro	Aberystwyth
Manceinion	40.70	parhaol	Aberystwyth



Ffigur 426: Prisiau diweddaru o Gaerdydd

Rydym bellach yn nodi ***Llundain*** fel y nôd dros dro gyda'r gost isaf o £68.10. Mae hyn yn cael ei osod i ***barhaol***, ac yn dod y ***nôd cyfredol***. Mae gan Lundain cysylltiad i'r nôd dros dro ***Dulyn***. Fodd bynnag, byddai'r pris i Ddulyn trwy Lundain fod yn ddrutach na'r gwerth a ddangosir eisoes, felly mae hyn yn cael ei anwybyddu.



Ffigur 427: Gwirio am ddiweddariadau pris o Lundain

Yr ydym yn awr yn adnabod ***Belfast*** fel y nôd dros dro gyda'r gost isaf o £97.39. Mae hyn bellach wedi ei osod i ***barhaol***, ac yn dod y ***nôd cyfredol***. Mae gan Belfast cysylltiad i'r nôd dros dro olaf, ***Dulyn***. Byddai'r pris i Ddulyn trwy Belfast fod yn ddrutach na'r gwerth a ddangosir eisoes, felly mae hyn yn cael ei anwybyddu.

NÔD	PRIS TEITHIO	STATWS	BLAENOROL
Aberystwyth	0	permanent	
Belfast*	97.39	permanent	Manchester
Caerdydd	49.70	permanent	Aberystwyth
Dulyn	108.50	temporary	Cardiff
Llundain	68.10	permanent	Aberystwyth
Manceinion	40.70	permanent	Aberystwyth

Rydym yn canfod ***Dulyn*** fel y nôd dros dro olaf. Mae hyn yn cael ei osod yn ***barhaol***, ac mae'r algorithm yn dod i ben. Rydym bellach wedi datrys y rhwydwaith ac wedi darganfod y prisiau isaf o Aberystwyth i bob un o'r nodau eraill. Wrth ddychwelyd at ein tasg wreiddiol o ddod o hyd i'r llwybr rhataf, gallwn weithio yn ôl o Ddulyn, gan ddefnyddio'r cofnodion yn y golofn ***flaenorol*** i ganfod pob nod ar hyd y llwybr:

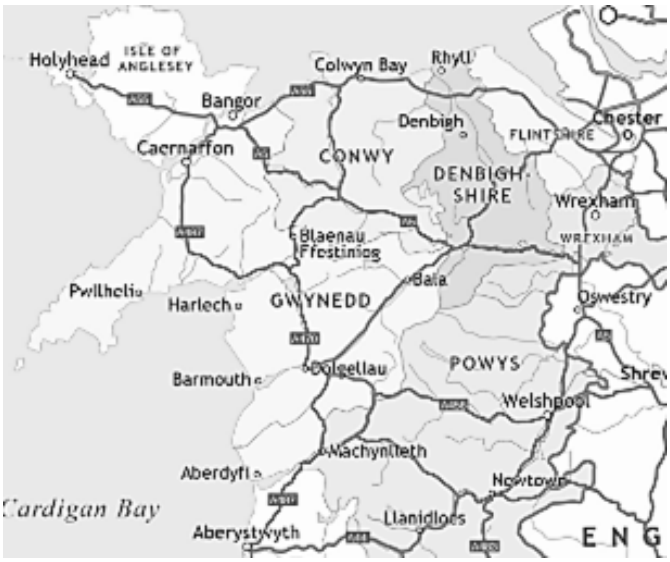
Mae Dulyn yn cael ei gyrraedd o Gaerdydd
Mae Caerdydd yn cael ei gyrraedd o Aberystwyth.

Dylai'r myfyriwr teithio ar y trên i Gaerdydd, yna cymerwch y cysylltiadau trên a fferi i Ddulyn.

Problem gwerthwr teithiol

Tasg gyffredin arall ar gyfer meddalwedd cynllunio llwybr yw dod o hyd i'r ffordd fyrraf neu cyflymaf i wneud taith o gwmpas cyfres o bwyntiau, yn ymweld â phob pwynt unwaith yn unig, ac yna dychwelyd i'r man cychwyn. Gelwir hyn yn **broblem gwerthwr teithiol**:

> Mae busnes yn Nolgellau yng Ngogledd Cymru yn cynhyrchu eitemau crefft. Dylai archebion yn cael eu dosbarthu i siopau yn: Aberystwyth, Caernarfon, Caergybi a'r Rhyl. Dewch o hyd i'r llwybr gorau ar gyfer y fan ddosbarthu i'w cymryd.



Ffigur 428:

Map o leoliadau Gogledd Cymru

Rydym yn dechrau drwy lunio tabl o'r pellteroedd mewn km rhwng y lleoliadau trosglwyddo:

	Dolgellau	Aberystwyth	Caernarfon	Caergybi	Rhyl
Dolgellau		54	70	115	89
Aberystwyth			125	170	139
Caernarfon				46	63
Caergybi					87

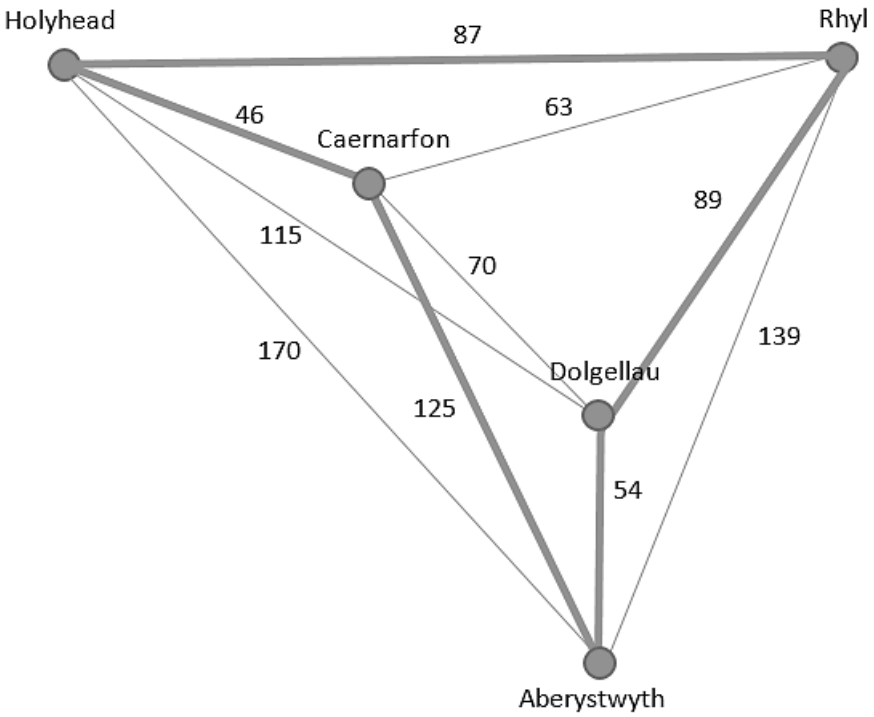
Gall darganfod llwybr posibl drwy ddefnyddio'r **algorithm cymydog agosaf**:

> Rydym yn dewis man cychwyn ar gyfer y daith dosbarthu. Y dref agosaf yn cael ei ddewis, ac yn gosod fel y pwynt nesaf ar lwybr y daith. O'r dref yr ydym wedi cyrraedd nawr, rydym yna dewiswch yr agosaf o'r trefi nad wedi cael ymweliad eto ac ychwanegu hyn at y llwybr. Mae'r broses yn parhau nes bod pob tref wedi ei ymwelwyd â hwy, yna byddwn yn dychwelyd yn uniongyrchol i'r man cychwyn i gwblhau'r daith gylchol.

Rydym yn dewis Dolgellau fel y man cychwyn ar gyfer y daith.

O Ddolgellau, y nôd agosaf yw Aberystwyth yn 54 km
O Aberystwyth, y nôd agosaf heb ei ymweld yw Caernarfon ar 125 km
O Gaernarfon, y nôd agosaf heb ei ymweld yw Caergybi ar 46 km
O Gaergybi, y nôd olaf heb ei ymweld yw'r Rhyl ar 87 km
Mae'r cysylltiad i ddychwelyd i Ddolgellau yw 89 km

Mae hyn yn gwneud cyfanswm bellter y daith o **401 km**. Mae'r llwybr ei ddangos yn ffigwr 429 isod:



Ffigur 429: Diagram rhwydwaith ar gyfer y lleoliadau dosbarthu

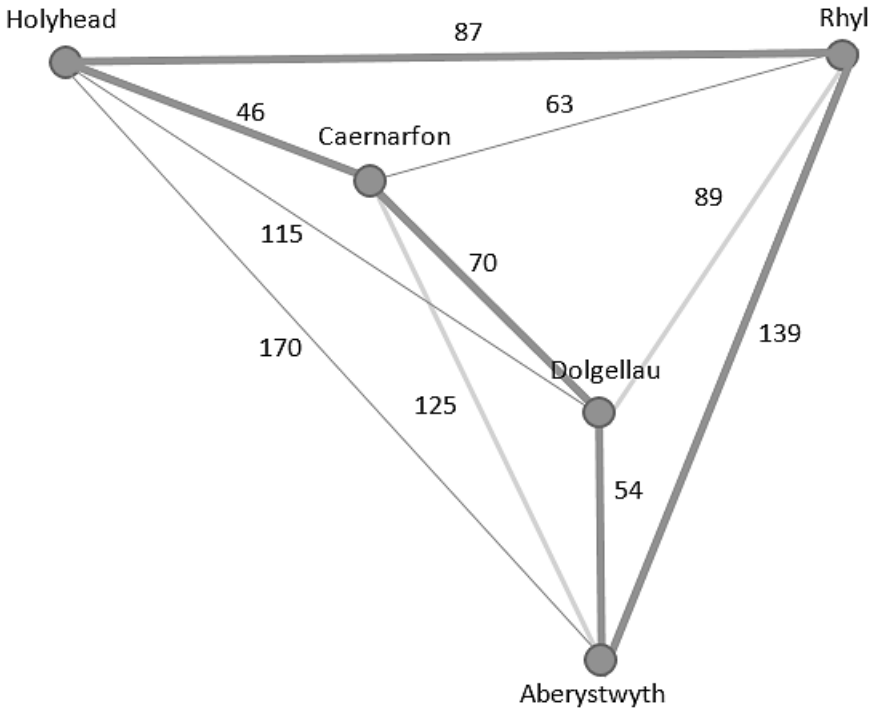
Mae problem gyda'r algorithm cymydog agosaf, yn wahanol i algorithm Dijkstra, yw dydy ei ddim yn gwarantu dod o hyd i'r ateb **gorau** ar gyfer y broblem. Gallwn archwilio hyn drwy geisio arbrawf:

Gan fod pob tref ei ymweld unwaith yn unig, ni ddylai'r llwybr byrraf yn dibynnu ar ble yn y ddolen i ni ddechrau.

Os ydym yn dewis Aberystwyth yn hytrach fel y man cychwyn:

O Aberystwyth, y nôd agosaf yw Dolgellau yn 54 km
O Ddolgellau, y nôd agosaf heb ei ymweld yw Caernarfon ar 70 km
O Gaernarfon, y nôd agosaf heb ei ymweld yw Caergybi ar 46 km
O Gaergybi, y nôd olaf heb ei ymweld yw'r Rhyl ar 87 km
Mae'r cysylltiad i ddychwelyd i Aberystwyth yw 139 km

Mae hyn yn rhoi cyfanswm pellter taith ychydig yn fyrrach o **396 km**. Mae'r llwybr ei ddangos yn ffigwr 430 isod.



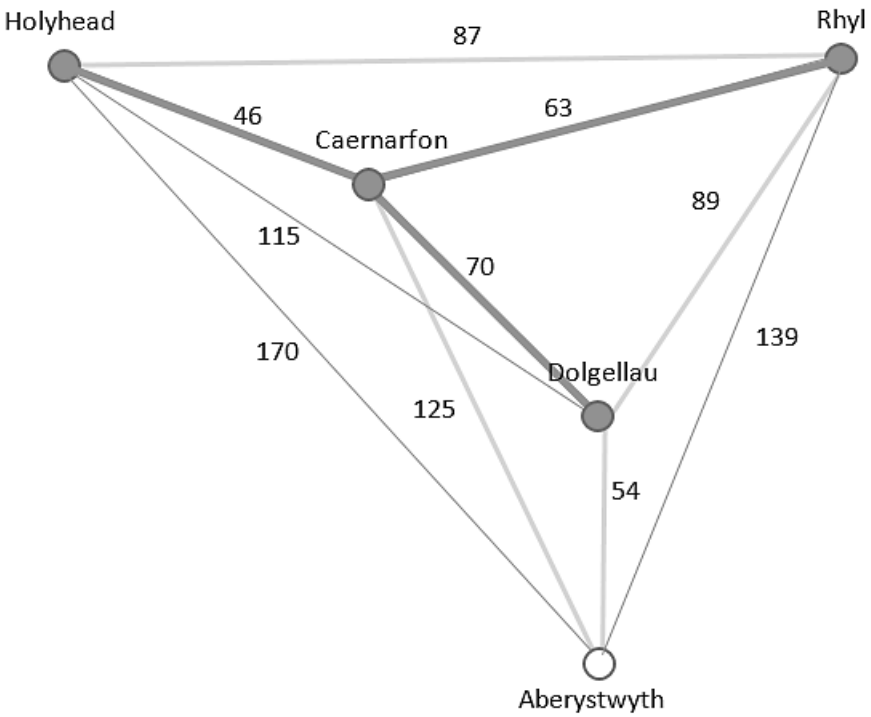
Ffigur 430: Llwybr a gynhyrchir gan algorithm y cymydog agosaf, gan ddechrau o Aberystwyth

Gall ansicrwydd yn y toddiant yn achosi anawsterau, yn enwedig ar gyfer rhwydweithiau mawr a chymhleth lle mae llawer o fannau cychwyn gwahanol yn bosibl. Mae ffordd, fodd bynnag, o ddarganfod terfyn is, o dan na all yr ateb fod. Er mwyn gwneud hyn:

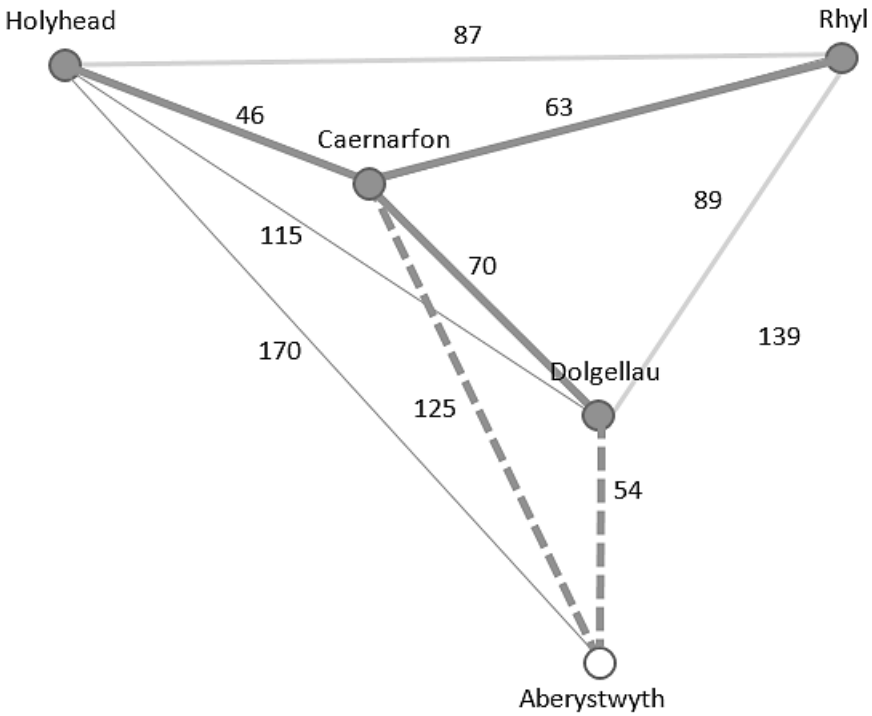
> Tynnwch un o'r nodau o'r rhwydwaith dros dro, er enghraifft **Aberystwyth**. Yna byddwn ddod o hyd i'r ffordd fyrraf o gysylltu'r pedwar nodau sy'n weddill. Gellir gwneud hyn drwy ddefnyddio pellteroedd cyswllt o 46 km, 63 km a 70 km fel y dangosir yn ffigwr 431.
>
> Ail-atodwch y nôd ar goll am Aberystwyth, ac yn ei gysylltu â'r rhwydwaith gan y ddau gysylltiad byrraf posibl o 54 km a 125 km, fel y dangosir yn ffigwr 432.

Yn y cam cyntaf, rydym yn cysylltu pob ond un o'r nodau gan ddefnyddio'r pellteroedd byrraf posibl. Byddai cylch di-dor o gwmpas nodau hyn yn rhoi'r o leiaf y pellter hwn, ac mae'n debyg i fod yn hirach. Yna cafodd y nôd ar goll ei hailgysylltu drwy ddefnyddio'r ddau bellter byrraf posibl. Byddai rhaid i'r nôd hwn cael dau gysylltiad gyda phellter o leiaf y rhain yn y cylched di-dor. Felly, gallwn ddweud gyda sicrwydd na allai unrhyw gylched wir yn cael cyfanswm o bellter byrrach na'r set hon o gysylltiadau. Cyfanswm hyd rydym wedi darganfod yw **358 km**.

Rydym yn gallu dweud bod y llwybr byrraf posibl o amgylch yr holl bwyntiau, gan ddychwelyd i'r dechrau, yn rhaid bod o leiaf **358 km**, ac yn llai na neu'n hafal i **396 km** yr ydym wedi cyfrifo yn gynharach.

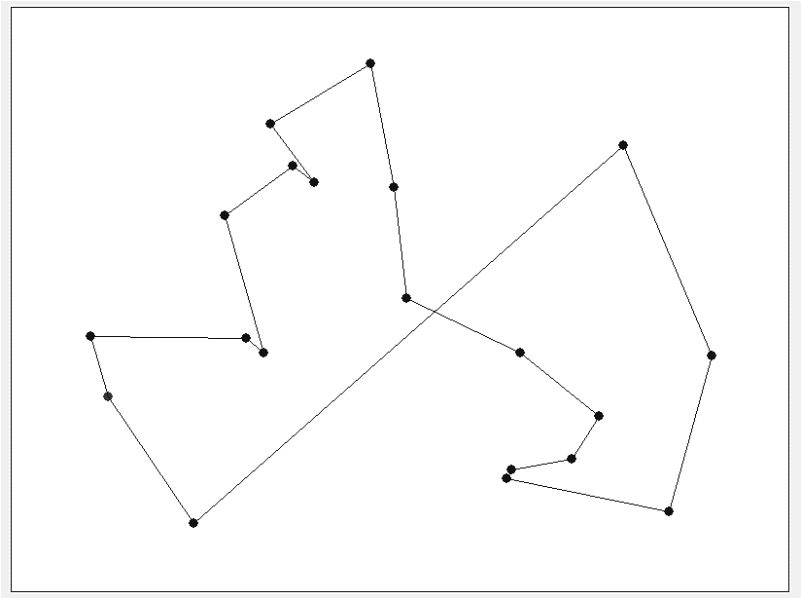


Ffigur 431: Rhwydwaith ar ôl diddymu Aberystwyth a chysylltu'r gweddill o nodau gyda'r pellteroedd byrraf.



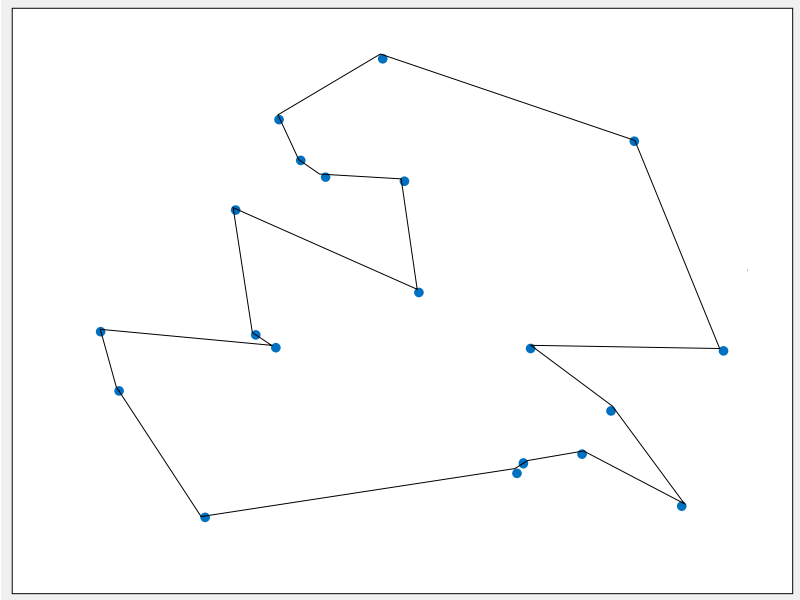
Ffigur 432: Rhwydwaith ar ôl ailosod a chysylltu'r nôd

Er bod algorithm y cymydog agosaf wedi rhoi ateb rhesymol i'r broblem syml hon, gall anawsterau yn digwydd gyda rhwydweithiau fwy. Nid yw'n anarferol i ddolenni ffigur- êt cael eu creu, fel yn ffigwr 433, gan arwain at gyfanswm fwy o bellter taith. Un ateb yw rhoi cynnig ar amrywiaeth o fannau cychwyn gwahanol ar gyfer y gylched, fel y gwnaethom yn gynharach, yn y gobaith o ganfod y llwybr byrraf. Fodd bynnag, gall hyn yn cymryd llawer o amser. Mae opsiwn gwell yw defnyddio algorithm optimeiddiaeth i wella'r canlyniad cychwynnol. Mae llwybr byrrach trwy'r un set o bwyntiau yn cael ei ddangos yn ffigur 434.



Ffigur 433:

Llwybr cychwynnol drwy set o bwyntiau a gynhyrchir gan yr algorithm cymydog agosaf

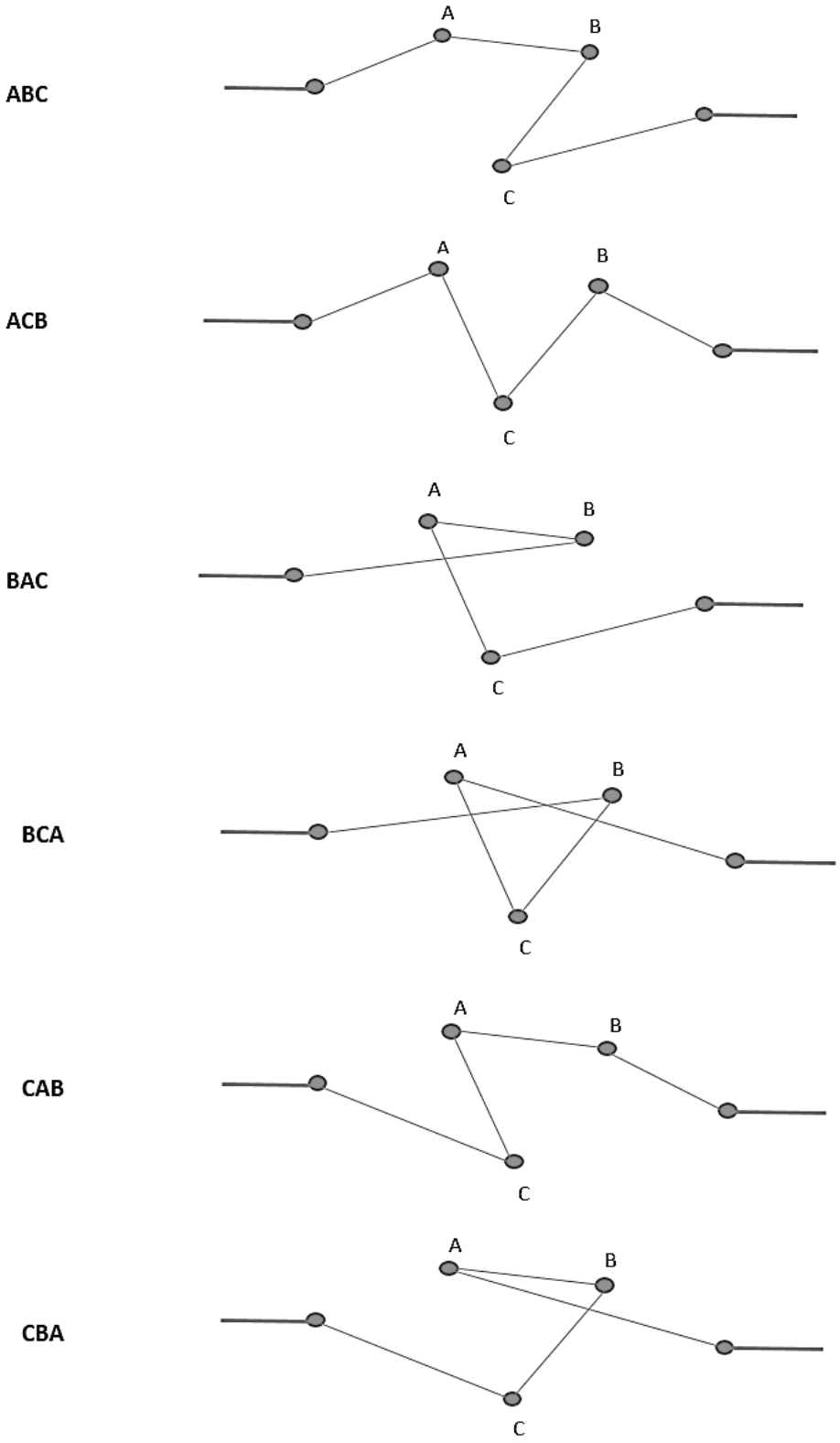


Ffigur 434:

Llwybr gwell trwy'r set o bwyntiau ar ôl optimeiddiaeth

Dull optimeiddiaeth syml yw cymryd pob grŵp o dri phwynt cyfagos ar hyd y llwybr, yna gwiriwch y set o gysylltiadau posibl rhwng y pwyntiau. Mae'r set fyrraf o gysylltiadau yn cael ei ddewis.

Pe bai tri phwynt yn cael eu labelu A, B ac C, yna mae cyfanswm o chwe dilyniant gwahanol yn bosibl:



Ffigur 407: Dilyniannau cysylltiad amgen ar gyfer tri phwynt A, B a C

I gynnal y weithdrefn optimeiddio, rydym yn dechrau ar bwynt cyntaf y llwybr. Yna, byddwn yn dewis y tri phwynt nesaf, ac yn eu galw A, B ac C. Y gwahanol ddilyniannau a ddangosir yn ffigur 435 uchod yna yn cael eu profi, a'r byrraf ei dethol. Yna, gall y dilyniant o'r pwyntiau yn y llwybr gwreiddiol yn cael ei aildrefnu os oes angen.

Yna, byddwn yn symud ymlaen un nôd, ac yn cynnal optimeiddio ar gyfer y tri phwynt nesaf A, B ac C. Mae'r weithdrefn hon yn cael ei ailadrodd nes bod y llwybr yn cael ei gwblhau ac rydym wedi dychwelyd i'r man cychwyn.

Efallai dal na fydd y llwybr y gorau y gallwn ganfod. Gall y dilyniant optimeiddio cyflawn yn cael ei wneud ar y llwybr gymaint o weithiau ag y bo angen, hyd nes y gellir dim gwelliant pellach yng nghyfanswm y pellter yn cael ei gyflawni.

Didoli

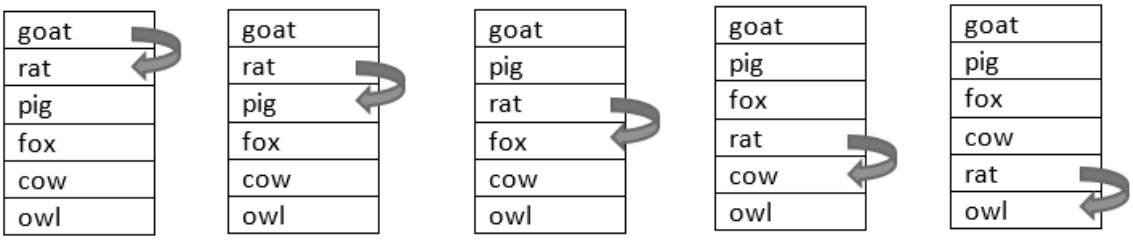
Gofyniad cyffredin mewn llawer o fathau o raglen gyfrifiadurol yw didoli data naill ai mewn trefn yr wyddor, trefn rifiadol neu drefn ddyddiad. Er enghraifft: gallai cronfa ddata o gwsmeriaid yn cael eu trefnu yn nhrefn yr wyddor yn ôl cyfenw, neu efallai taenlen o ganlyniadau arholiadau myfyrwyr yn cael eu didoli mewn trefn rifiadol o'r marciau a ddyfarnwyd.

Er bod meddalwedd cyfrifiadurol pwrpas cyffredin ar gael ar gyfer cynnal y rhan fwyaf o dasgau gweinyddol, efallai y bydd angen i fusnes neu sefydliad ddatblygu ei meddalwedd arbenigol eu hunain at ddibenion penodol. Rhaglenni yn debygol iawn o fod â gofyniad i ddidoli data ar ryw gam. Bydd sgiliau rhifedd o gydnabyddiaeth patrwm a datrys problemau yn bwysig i'r rhaglenwyr sy'n datblygu algorithmau didoli ar gyfer y meddalwedd hwn.

Dull ddidoli syml yw **trefniad swigen**. Er mwyn dangos sut mae hyn yn gweithio, byddwn yn cymryd y gyfres o eiriau Saesneg:

goat rat pig fox cow owl

ac yn ceisio i'w rhoi mewn trefn yr wyddor. Rydym yn symud i lawr y rhestr, gan gymharu pob pâr o eiriau yn eu tro. Os bydd y drefn unrhyw bâr o air yn anghywir, mae'r geiriau yn cael eu cyfnewid.



Ffigur 436: Pàs gyntaf trwy restr o eiriau yn ystod y trefniad swigen

Wrth archwilio'r dilyniant yn ffigwr 436 uchod, rydym yn gweld bod:

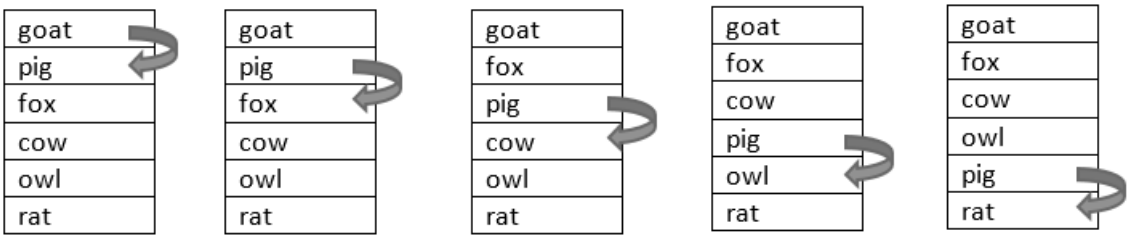
goat a **rat** eisoes yn nhrefn yr wyddor gywir
rat a **pig** mewn trefn anghywir fel eu bod yn eu cyfnewid
rat a **fox** mewn trefn anghywir fel eu bod yn eu cyfnewid ...

Ar ddiwedd y pàs cyntaf drwy'r holl ddata, mae gennym y dilyniant:

goat
pig
fox
cow
owl
rat

Nid yw hyn yn drefn yr wyddor gywir eto, ond mae geiriau agosach at gychwyn y dilyniant wedi ' bybylu i fyny' tuag at frig y rhestr – felly'r enw **trefniad swigen**.

Rydym yn defnyddio'r dilyniant wedi ei haddasu fel man cychwyn ar gyfer pàs arall drwy'r data:

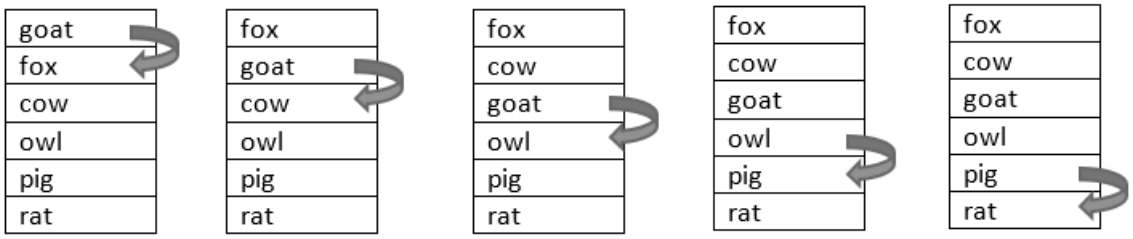


Ffigur 437: Ail bàs drwy'r rhestr o eiriau yn ystod y trefniad swigen

Erbyn hyn mae gennym y dilyniant:

goat
fox
cow
owl
pig
rat

Mae hyn yn dal i fod yn nhrefn y wyddor anghywir, felly mae pàs arall yn cael ei wneud:



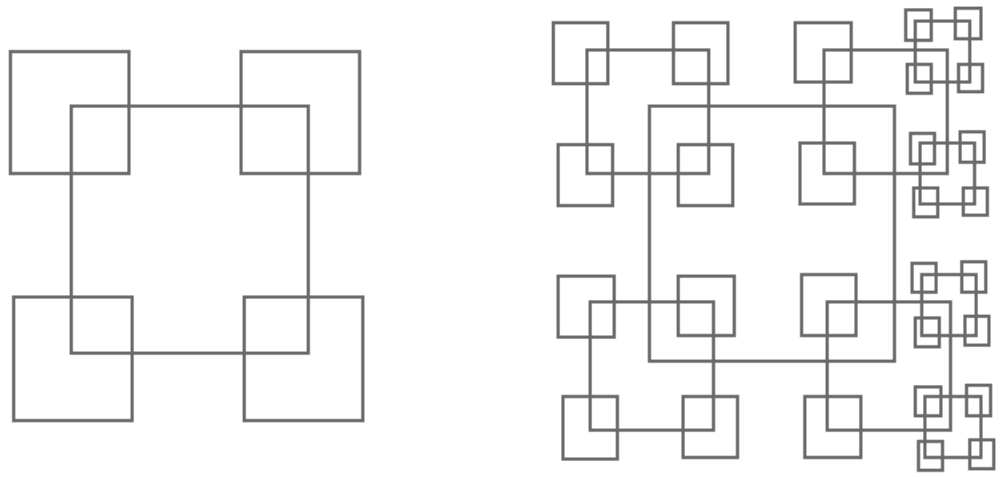
Ffigur 438: Trydydd pàs drwy'r rhestr o eiriau yn ystod y trefniad swigen

Mae'r rhestr yn dal i fod yn drefn ychydig yn anghywir. Bydd angen un pàs ychwanegol drwy'r data i ddod a'r gair **cow** i frig y rhestr. Mae nifer fawr o gymariaethau wedi bod yn angenrheidiol ac nid yw'r algorithm yn arbennig o effeithlon. Os oes rhaid i lawer o eitemau yn cael eu trefnu yn ôl dull hwn, byddai'r rhaglen yn araf. Mae angen dulliau didoli gwell ar gyfer setiau mawr o ddata.

Mae gwell dull didoli, yr **algorithm quicksort**, yn gwneud defnydd o dechneg o **ddychweliad**.

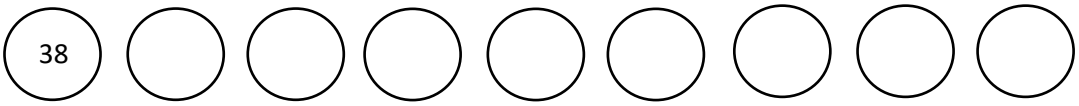
Mae dychweliad yn golygu cynnal fersiwn o dasg *o fewn y dasg ei hun*. Enghraifft syml yw cynhyrchu patrwm geometrig drwy ddefnyddio dychweliad.

> Rydym yn dechrau drwy luniadu sgwâr.
> Yna mae pedwar sgwariau llai yn cael eu lluniadu gyda'u canolau ar bob un o'r corneli'r sgwâr mwy.
> Pellach y gellir sgwariau llai wedyn yn cael ei ychwanegu yn y corneli o bob un o'r sgwariau bychain hyn, gan gynhyrchu patrwm o unrhyw ddyfnder a ddewiswyd.



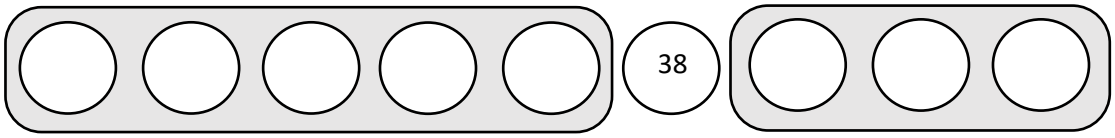
Ffigur 439: Datblygu patrwm dychweliad geometrig

Mae'r **algorithm quicksort** yn dechrau gyda set o ddata heb eu didoli. Un eitem data, yn aml y cyntaf yn y dilyniant, yn cael ei ddewis:



Ffigur 440: Colyn wedi ei dewis fel yr eitem gyntaf yn y rhestr heb eu didoli

Mae'r rhaglen wedyn yn cymharu eitem cyfeirio, a adwaenir fel y **colyn**, i bob un o'r eitemau eraill yn y set ddata. Mae pob eitem gyda gwerthoedd is yn cael eu symud o flaen y colyn, tra bod yr holl eitemau â gwerthoedd uwch yn parhau i fod ar ôl y colyn.



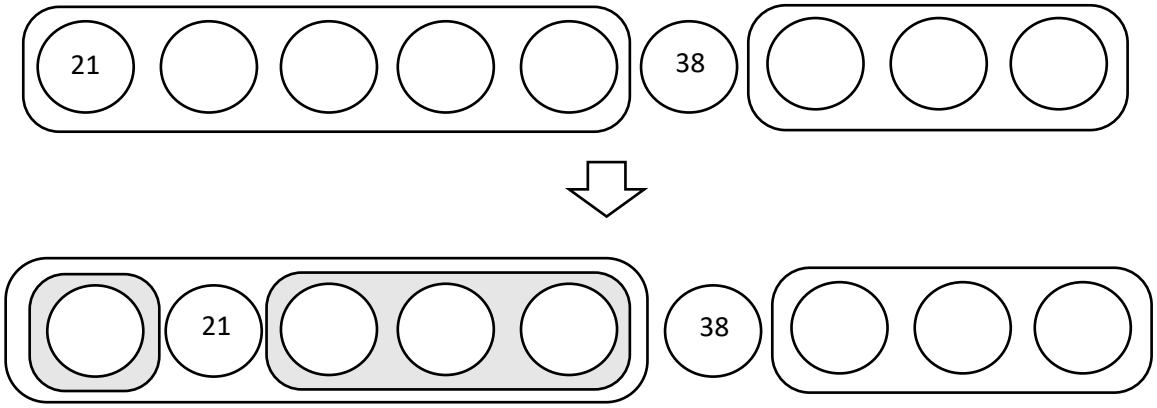
is-rhestr heb ei ddidoli, ond mae bob eitem â gwerth llai na 38

is-rhestr heb ei ddidoli, ond mae bob eitem â gwerth uwch na 38

Ffigur 441: Rhaniad o'r rhestr wreiddiol yn ddwy is-restr

Rydym bellach wedi rhannu'r rhestr wreiddiol yn ddwy is-restr heb eu didoli, wedi'u gwahanu gan yr eitem colyn. Rydym yn gwybod bod y colyn bellach yn ei safle cywir, gan fod unrhyw ad-drefnu sy'n angenrheidiol yn yr is-restrau ni fydd yn effeithio ar y colyn.

Mae'r weithdrefn didoli nawr yn cael ei ailadrodd yn **dychweliadol** ar bob un o'r is-rhestrau. Mae gwerthoedd colyn newydd yn cael eu defnyddio i wneud cymariaethau i greu rhagor o is-restrau:



Ffigur 442: Didoli pellach o is-rhestr drwy ddychweliad

Mae dychweliad yn parhau nes bod yr holl is-restrau wedi cael eu lleihau i eitemau unigol. Ar y pwynt hwn, bydd y set o ddata wedi cael ei threfnu yn llawn.

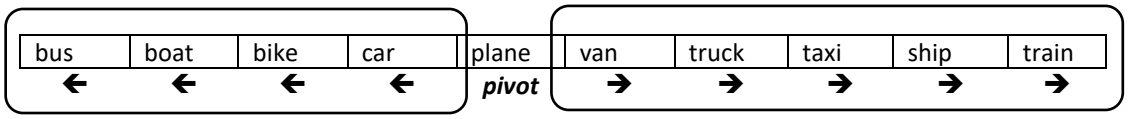
Er mwyn dangos yr algorithm quicksort yn gweithredu, byddwn yn ystyried y gyfres o eiriau Saesneg ar hap am ddulliau o drafnidiaeth. Mae angen didoli'r geiriau hyn yn nhrefn yr wyddor.

plane	van	truck	bus	boat	taxi	ship	bike	car	train

Byddwn yn dewis y gair '***plane***' ar ddechrau'r dilyniant fel colyn. Yna byddwn yn penderfynu a fyddai pob un o'r geiriau eraill yn dod o flaen (🡐) neu ar ôl (🡒) y colyn yn nhrefn yr wyddor.

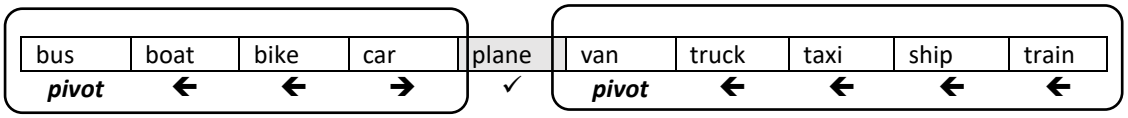
plane	van	truck	bus	boat	taxi	ship	bike	car	train
pivot	🡒	🡒	🡐	🡐	🡒	🡒	🡐	🡐	🡒

Rydym yn gwneud copi newydd o'r data, gan roi'r holl eitemau cyn gwerth y colyn ar y chwith, ac wedyn y colyn ei hun, yna'r holl eitemau ar ôl y colyn ar y dde.



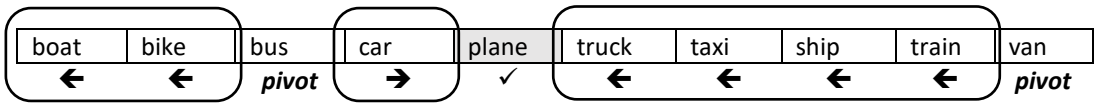
Er nad yw grwpiau o ddata i'r chwith ac i'r dde o'r colyn yn cael eu trefnu eto, mae'n rhaid i'r colyn ei hun fod yn y safle cywir yn y dilyniant. Ni all unrhyw didoli pellach yn newid ei lleoliad.

Mae'r broses nawr yn cael ei ailadrodd ar gyfer y grwpiau o ddata heb ei ddidoli ar y chwith a de o'r colyn. Gwerthoedd gymhariaeth newydd bydd '***bus***' a '***van***'.



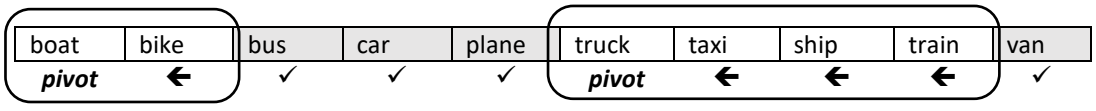
O fewn pob grŵp heb eu didoli, byddwn yn penderfynu dylai pob gair fod cyn neu ar ôl gwerth y colyn.

Mae'r geiriau ym mhob grŵp heb ei ddidoli yn cael eu had-drefnu unwaith eto. Rhestrir geiriau cyn i'r colyn yn gyntaf, yna'r colyn ei hun, ac wedyn y geiriau sy'n dod ar ôl gwerth y colyn yn nhrefn yr wyddor.

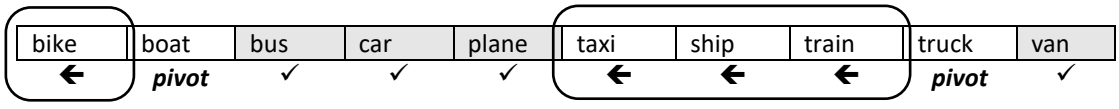


Mae'r gwerthoedd colyn '***bus***' a '***van***' nawr mewn lleoliadau cywir yn y dilyniant. Fodd bynnag, rhaid i '***car***' hefyd fod mewn lleoliad cywir gan ei fod eitem yn unig na ellir eu cyfnewid ag unrhyw air arall.

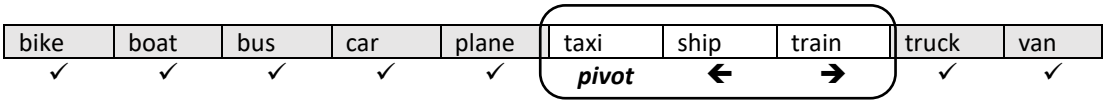
Mae'r trefniad yn parhau drwy greu gwerthoedd cymhariaeth colyn ar gyfer y grwpiau heb eu didoli eto.



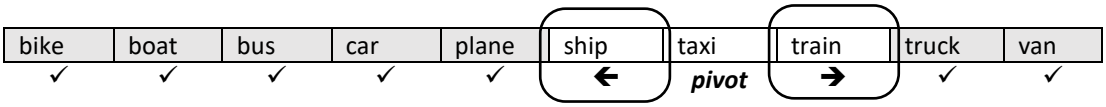
Aildrefnu eitemau'r gweddill y data yn rhoi:



Mae'r eitemau colyn '***boat***' a '***truck***' bellach mewn lleoliadau cywir. Mae'n rhaid i'r eitem sengl '***bike'*** hefyd fod yn gywir. Mae trefnu yn parhau gydag un grŵp yn weddill o eitemau.



Mae'r ad-drefnu terfynol yn cwblhau'r didoli.



Gallwn weld bod fersiynau o'r dull didoli yn cael eu cynnal ***o fewn ei hun***, fel y gall y broblem ei datrys drwy ddychweliad.

Dywedwyd yn gynharach fod yr algorithm quicksort yn ddull didoli cyflymach a mwy effeithlon na'r algorithm trefniad swigen. Gadewch i ni ystyried pam fod hyn yn wir.

Mae effeithlonrwydd, neu ***radd***, o algorithmau wedi ei bennu yn aml drwy gyfrwng **nodiant O-mawr**. Mae hon yn ffordd o ddisgrifio effaith ar amser didoli os yw maint y data yn cynyddu.

Gadewch i ni yn gyntaf edrych ar yr algorithm **trefniad swigen**. Heb boeni gormod am yr union fecanwaith didoli, gallwn nodi dau gam cyffredinol sydd eu hangen i roi'r eitemau data i mewn i'w leoliadau cywir mewn rhestr:

- rhaid i bob eitem o ddata yn cael eu dewis
- rhaid i'r eitem o ddata cael ei gymharu â phob un o'r eitemau data eraill

Bydd pob un o'r prosesau hyn yn cymryd rhywfaint o amser prosesu, a bydd pob un yn dibynnu ar y nifer o eitemau data **n** sydd yn bresennol. Os bydd y nifer o eitemau data yn cael ei dyblu, yna bydd y ddwy broses yn cymryd dwywaith yr amser. Bydd effaith cyflawn gwneud y proses o ddidoli yn bedair gwaith yn arafach. Rydym yn dweud bod trefn y dull didoli yw:

$$O(n^2)$$

sy'n golygu bod yr amser cyflawn ar gyfer yr algorithm mewn cyfrannedd â sgwâr o nifer yr eitemau data.

Gan droi nawr at y **quicksort**, gallwn ddweud eto bod dwy broses yn cymryd rhan:

- rhaid i bob eitem o ddata yn cael eu dewis
- rhaid i'r eitem o ddata yn cael ei gymharu â phob un o'r eitemau eraill yn yr is-restr.

Fodd bynnag, cynydd yn nifer yr eitemau data yn ei gwneud dim ond newid bach i faint cyfartalog yr is-rhestrau. Mae dyblu nifer yr eitemau data yn unig yn ychwanegu un rhestr mwy o faint ar y lefel uchaf y dychweliad, gyda'r holl is-rhestrau isod hwn sy'n weddill o'r un maint. Byddai cynyddu faint o ddata gan 256 o weithiau ond yn ychwanegu wyth lefel ychwanegol i'r goeden dychweliad, felly ni fyddai fawr ddim newid i hyd cyfartalog yr is-rhestrau sy'n cael eu didoli.

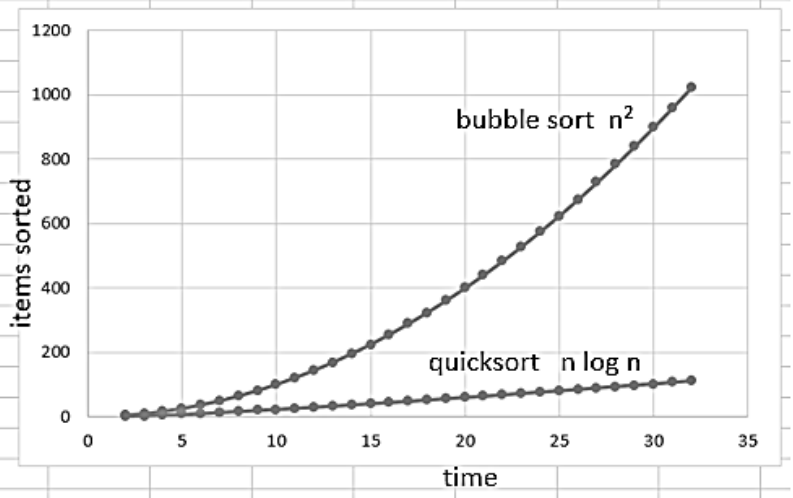
Fe'i ceir bod trefn yr algorithm quicksort yw:

$$O(n \log n)$$

Bydd yr amser ddidoli yn gymesur â nifer yr eitemau data **n** luosi gan logarithm y rhif hwn.

Gallwn weld pam mae quicksort yn gyflymach na threfniad swigen drwy blotio graffiau o'r ddau ffwythiant n^2 a n log n:

N	N squared	N ln N
2	4	1.38629436
3	9	3.29583687
4	16	5.54517744
5	25	8.04718956
6	36	10.7505568
7	49	13.621371
8	64	16.6355323
9	81	19.7750212
10	100	23.0258509
11	121	26.376848
12	144	29.8188798
13	169	33.3443416
14	196	36.9468026
15	225	40.620753
16	256	44.3614196
17	289	48.1646268
18	324	52.0266916
19	361	55.9443406
20	400	59.9146455
21	441	63.9349712
22	484	68.002934
23	529	72.116367



Ffigur 443: Graffiau o n^2 a n log n

Er bod yr amser didoli ar gyfer quicksort yn codi mewn modd bron llinellol wrth i'r nifer o eitemau data yn cynyddu, yr amser didoli ar gyfer y ***trefniad swigen*** yn cynyddu yn fwy serth. Ar gyfer symiau mawr o ddata, bydd fantais fawr wrth ddefnyddio'r dull quicksort. Fodd bynnag, gyda symiau bach o ddata nad oes fawr o wahaniaeth mewn cyflymder, ac efallai y byddai'n well gan raglenwyr ddefnyddio'r algorithm trefniad swigen sy'n symlach wrth ei rhaglennu.

Amgryptio

Mae'n aml yn bwysig i amgryptio data cyfrifiadurol, er enghraifft, pan fydd yn cael ei anfon dros rwydwaith a allai gael ei rhyng-gipio gan berson heb awdurdod. Un maes pwysig o fathemateg yn canolbwyntio ar ddatblygu dulliau diogel ac effeithlon o amgryptio data, ac yn aml yn cynnwys datblygu algorithmau cymhleth.

Mae data a drosglwyddir dros rwydwaith neu ei storio ar gyfryngau digidol fel arfer yn gyfres o rifau bach sy'n cynrychioli llythrennau'r wyddor, digidau neu nodau bysellfwrdd arall. Mae system a ddefnyddir yn gyffredin yw Cod Safonol Americanaidd ar gyfer Cyfnewid Gwybodaeth (ASCII) sy'n cynrychioli'r cymeriadau ar fysellfwrdd Saesneg safonol gan niferoedd yn yr ystod 0 i 255. Yn y system hon, mae cymeriad 'A' yn cael y gwerth 65, ' B ' yn 66,' C ' yn 67, ac yn y blaen ...

Mae seiffrau yn gweithio ar yr egwyddor o ffwythiannau gwrthdro. Rydym yn gyfarwydd â'r syniad o rai gweithrediadau mathemategol yn wrthwyneb rai eraill. Er enghraifft, gallwn wrthdroi ychwanegiad drwy tynnu:

$$65 + 6 = 71 \qquad 71 - 6 = 65$$

Gallai cod ASCII 65 yn cynrychioli'r llythyren A yn cael ei amgryptio drwy ychwanegu 6, fel bod y cod a drosglwyddir yn 71, yn cynrychioli'r llythyr G. Gallai pob cod nodau eraill yn y neges yn cael ei hamgryptio yn yr un modd. Er enghraifft, mae'r neges:

HELLO

yn dod yn **NKRRU**

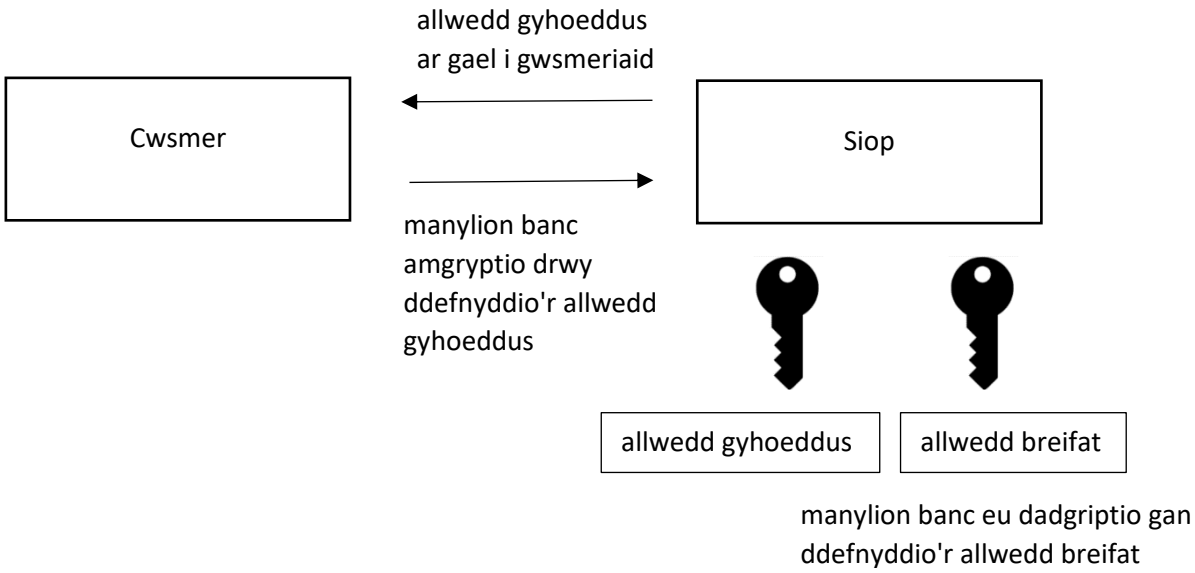
Os bydd gwerth yr **allwedd** amgryptio o 6 yn ei adnabod, gall y neges yn cael ei dadgryptio drwy dynnu hyn o'r gwerthoedd cod ASCII a dderbyniwyd. Mae'r dull hwn o symud llythrennau'r wyddor gerbron gan nifer benodol o leoedd yn cael ei ddefnyddio yng nghyfnod y Rhufeiniaid, ac yn cael ei adnabod fel y **Caesar Cypher**. Fodd bynnag, gall y seiffr yn hawdd ei dorri drwy ddadansoddi pa mor aml bod llythrennau yn y neges: nodau sy'n digwydd mwyaf aml yn y neges wedi ei amgryptio yn debygol o fod llythrennau cyffredin fel E, T neu A.

Mae dulliau mwy cymhleth o amgryptio wedi cael eu datblygu, megis yr Enigma Cypher a ddefnyddiwyd yn ystod yr Ail Ryfel Byd oedd yn dibynnu ar beiriannau i amgryptio gyda ffwythiant mathemategol cymhleth, yna dadgryptio trwy gyfrwng y ffwythiant gwrthdro. Er bod llawer mwy diogel, roedd anhawster yn dal i fodoli gyda'r system hon; roedd angen trosglwyddo manylion am osodiadau allweddi ymlaen llaw oddi wrth yr anfonwr at y derbynnydd fel bod y wybodaeth angenrheidiol ar gael i ddadgryptio'r neges. Gallai hyn beri risg diogelwch os yw'r allweddi yn syrthio i'r dwylo anghywir.

Digwyddodd lwyddiant mawr pan gafwyd dulliau eu darganfod ar gyfer **amgryptio allweddi dwbl**. Yn y system hon, un allwedd yn cael ei ddefnyddio i amgryptio'r neges, yna allwedd wahanol yn cael ei ddefnyddio ar gyfer dadgriptio. Hyd yn oed os yw'r allwedd amgryptio yn hysbys, ni all yr allwedd dadgriptio yn cael ei gyfrifo ac mae'r neges yn parhau i fod yn ddiogel.

Fel enghraifft, ystyried cwsmer sy'n prynu nwyddau o siop ar-lein, ac yna talu gyda cherdyn credyd. Mae'n bwysig bod manylion cerdyn y cwsmer eu trosglwyddir yn ddiogel dros y rhyngrwyd at y siop. Byddai amgryptio allweddol sengl fod yn anaddas, gan y byddai anfon yr allwedd i'r cwsmer fod yn risg diogelwch. Os byddai'r allwedd ei rhyng-gipio gan droseddwr, gallai hyn wedyn yn cael ei ddefnyddio i ddadgryptio'r manylion cerdyn credyd.

Yn y system allwedd ddwbl, mae **pâr** o allweddi yn cael ei greu. Un, a elwir yr **allwedd gyhoeddus**, yn cael ei wneud ar gael yn rhydd fel rhan o dudalen we'r siop. Mae'r dudalen gwefan yn defnyddio hyn i amgryptio manylion banc pan fydd y cwsmer yn anfon eu harcheb. Nid yw gwybodaeth o'r allwedd gyhoeddus yn ei gwneud yn bosibl i ddadgryptio'r neges. Gall hyn ond yn cael ei wneud gan ddefnyddio aelod arall o'r pâr o allweddi, yr **allwedd breifat**, sy'n cael ei chadw'n ddiogel yn y siop.



Ffigur 444: Defnydd o system amgryptio allweddi dwbl

Mae llwyddiant y dull allweddi dwbl yn dibynnu ar ddod o hyd i ffwythiant mathemategol ni ellir ei benderfynu gan y wybodaeth sydd ar gael yn yr allwedd gyhoeddus y mae eu gwrthdro. Mae'r ffwythiant a gwrthdro a ddewiswyd yn rhai braidd yn anarferol. Mae amgryptio yn cael ei wneud drwy ddefnyddio ffwythiant **modwlws**.

> Mae'r gwerth heb ei amgryptio, megis y cod ASCII 65 ar gyfer 'A', yn cael yr enw **m**.
>
> Mae'r gwerth wedi'i amgryptio, a fydd yn cael ei anfon yn ddiogel dros y rhyngrwyd, yn cael yr enw **c**.
>
> Mae'r ddau rif, a elwir yn **e** a **n**, yn cael eu defnyddio yn ystod y broses amgryptio.

Yna caiff y gwerth **m** ei amgryptio i gynhyrchu **c** gan ddefnyddio'r hafaliad:

$$(m^e)\ MOD\ n = c$$

Mae'r gwerth **m** ei godi i bŵer **e**, yna dod o hyd i'r gweddill pan fydd y rhif **n** yn cael ei dynnu oddi wrtho gymaint o weithiau ag y bo modd.

Er enghraifft, tybied bod y cod ASCII 65 yn cael ei drosglwyddo. Gallai unrhyw werthoedd addas yn cael eu dewis ar gyfer y niferoedd **e** a **n** a fydd yn amgryptio'r data.

Byddwn yn dewis **e** = 3 a **n** = 143.

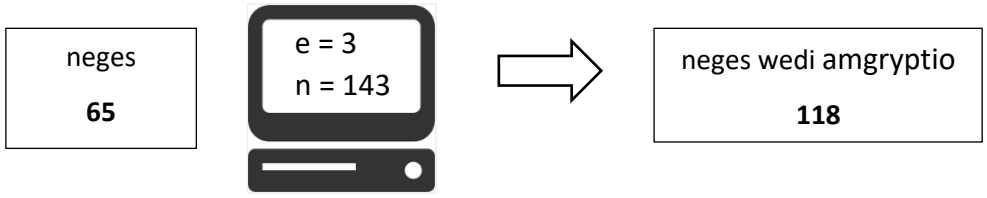
Cyflawni'r camau'r amgryptio:

$$(65^3)\ MOD\ n = c$$

$$493039\ MOD\ n = c$$

$$493039\ MOD\ 143 = 118$$

65 wedi ciwbio yw 493039. Tynnu 143 dro ar ôl tro o 493,039 yn gadael rhelyw o 118. Dyma'r gwerth amgryptio a fydd yn cael ei drosglwyddo.



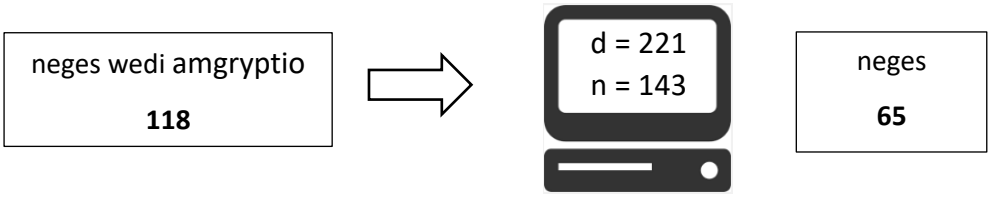
Ffigur 445: Amgryptio gan ddefnyddio gwerthoedd allwedd **e** ac **n**

Rydym bellach angen ffwythiant gwrthdro sy'n gallu newid 118 yn ôl at y gwerth gwreiddiol:

$$\left(c^d\right)\ MOD\ n = m$$

lle mae **c** y gwerth wedi'i amgryptio ac **m** y neges wreiddiol. Mae **n** yr un gwerth modwlws sy'n ei ddefnyddio yn ystod yr amgryptio.

Hefyd, mae angen gwerth arall **d**. Am resymau fydd yn cael eu hesbonio yn fuan, mae gwerth addas ar gyfer d yw 221.



Ffigur 446: Dadgriptio gan ddefnyddio gwerthoedd allwedd **d** ac **n**

I weithredu'r system amgryptio allweddi dwbl, byddai angen i'r cwsmer yn yr enghraifft siop i adnabod y gwerthoedd amgryptio **e** ac **n**. Gall y rhain fod ar gael i'r cyhoedd. Fodd bynnag, ni all y neges yn cael ei dadgriptio oni bod gwerth **d** ei adnabod hefyd. Byddai hyn yn cael ei chadw'n ddiogel ar system gyfrifiadurol y siop ac nid ei datgelu i bobl heb awdurdod.

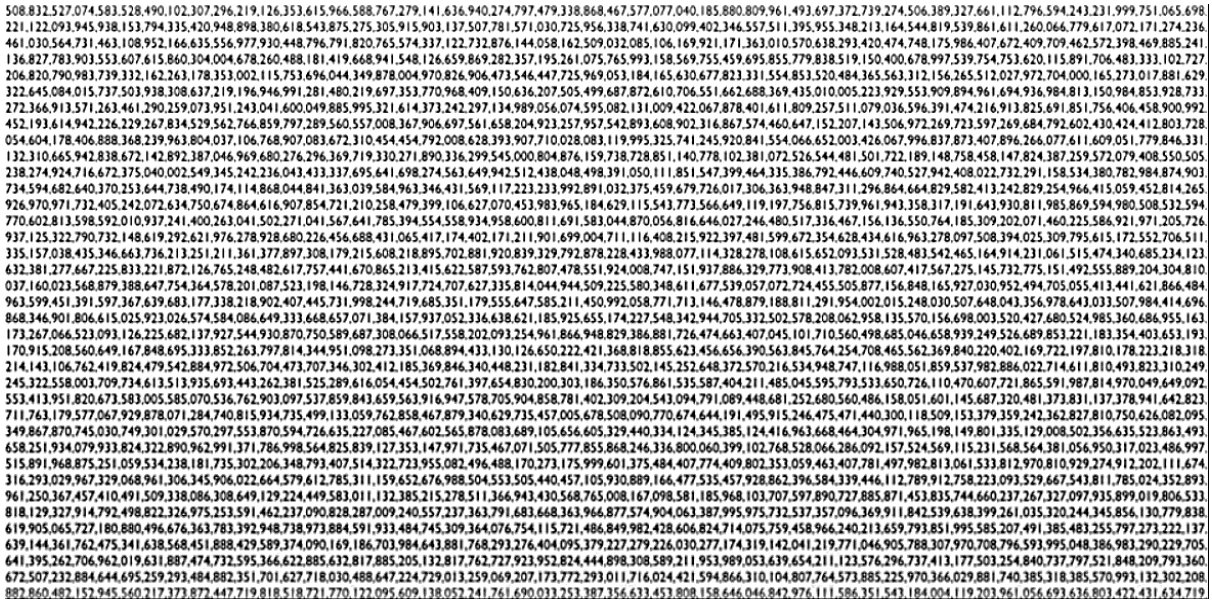
Mae diogelwch y dull amgryptio allweddi dwbl yn dibynnu ar ei fod yn amhosibl, neu'n anodd iawn o fewn amser rhesymol, i ddod o hyd i werth **d** pan mai dim ond **e** ac **n** eu nabod.

I weithredu amgryptio allweddi dwbl, mae arnom angen ffordd o gynhyrchu cyfres o werthoedd **e**, **n** a **d** y gellir eu defnyddio yn yr allweddi cyhoeddus a phreifat. Yn ymarferol, mae **n** yn cael ei ddewis i fod nifer fawr iawn sy'n cael ei gynhyrchu drwy luosi dau rif cysefin mawr y byddwn yn galw **p** a **q**. Er enghraifft, efallai y byddwn yn dewis:

$$p = 1889, \quad q = 3547$$

$$n = 1889 \times 3547 = 6{,}700{,}283$$

Dim ond gan wybod y lluoswm **n** yn ei gwneud yn anodd iawn dod o hyd i **p** a **q**. Mae systemau real yn defnyddio gwerthoedd hyd yn oed yn fwy, felly bod y dasg effeithiol yn amhosibl. Er enghraifft, mae'r gwerth a gynhyrchwyd ar ôl lluosi dau nifer mawr cysefin allai fod:



Ffigur 447: Enghraifft o werth amgryptio mawr diogel a gynhyrchwyd drwy luosi rhifau cysefin (Vance, 2014)

Byddai'n cymryd amser hir i ddod o hyd i'r ddau rif cysefin sy'n lluosi i roi canlyniad hwn! Mae arwyddocâd yw y gall y gwerth dadgriptio **d** i'w gael dim ond os ydym yn gwybod y ddau rif cysefin **p** a **q** a ddefnyddiwyd i gyfrifo **n**.

Mae'r dull a ddefnyddir i ddod o hyd i **d** o **p** a **q** yw'r **Algorithm Ewclidaidd Estynedig**. Mae'r weithdrefn yn eithaf cymhleth, ond gellir eu darlunio gan weithio drwy esiampl:

- Rydym yn dechrau drwy ddewis dau rif cysefin mawr iawn **p** a **q**. Er mwyn cadw pethau'n syml yn yr enghraifft hon, byddwn yn pennu:

  p = 5, q = 11

- Rydym bellach yn cyfrifo gwerth **φ** a elwir yn **ffwythiant totient Euler**. Gwneir hyn drwy dynnu 1 o bob un o'r p a q, yna lluosi'r canlyniad:

$$\varphi = (5-1)(11-1) = 40$$

- Rydym bellach yn dewis gwerth am yr allwedd amgryptio **e**. Mae'n bwysig nad yw hyn yn rhannu yn ffactor cyffredin gyda **φ**. Byddai gwerth o 7 yn addas ar gyfer **e**, ers nid yw 7 yn rhannu yn 40.
- Mae'r gwerth amgryptio arall **n** yn lluoswm **p** a **q**:

$$n = 5 \times 11 = 55$$

Nawr mae angen yn unig i ddod o hyd i'r allwedd dadgriptio d gan ddefnyddio'r Algorithm Ewclidaidd Estynedig. Er na fydd unrhyw un o'r camau'r algorithm yn cynnwys mathemateg anodd, gall y dilyniant fod yn anodd ei gofio. Yn ymarferol, byddai gwerthoedd allweddi yn cael ei gynhyrchu gan raglen gyfrifiadurol sydd wedi'i sefydlu i wneud y dilyniant cywir o gamau.

Rydym yn dechrau drwy ysgrifennu gwerth **φ** ar frig ddwy golofn. O dan hyn yn y golofn gyntaf yw gwerth **e**, ac yn yr ail golofn ei ysgrifennu 1:

40	40
7	1

Rydym bellach yn rhannu **φ** gan **e**, yn ysgrifennu dim ond rhif cyfan y canlyniad ac anwybyddu unrhyw relyw.

40	40
7	1
5	

Rydym yn lluosi canlyniad hwn gan bob un o'r gwerthoedd yr ail res:

40	40
7	1
5 * 7 = 35	5 * 1 = 5

Yna mae'r cyfansymiau hyn yn cael eu tynnu o'r gwerthoedd ar y rhes uchaf:

40	40
7	1
(40 – 35) = 5	(40 – 5) = 35

Mae'r canlyniadau hyn nawr yn ffurfio trydedd res y tabl:

40	40
7	1
5	35

Gall rhes gyntaf y tabl nawr yn cael ei dileu:

7	1
5	35

Mae'r camau a wnaed uchod nawr yn cael eu hailadrodd:

Rydym yn rhannu'r gwerth colofn chwith y rhes gyntaf gan werth colofn chwith yr ail res, yn ysgrifennu'r canlyniad fel rhif cyfan ac yn anwybyddu unrhyw relyw.

7	1
5	35
1	

Rydym yn lluosi'r canlyniad hwn gan bob un o'r gwerthoedd ail res:

7	1
5	35
1 * 5 = 5	1 * 35 = 35

Yna mae'r cyfansymiau hyn yn cael eu tynnu o'r gwerthoedd ar y rhes uchaf:

7	1
5	35
(7 – 5) = 2	(1-35) = -34

Gellir derbyn dim ond canlyniadau positif. Mae gwerth negyddol -34 cael ei newid i werth positif drwy ychwanegu **φ**:

$$-34 + 40 = 6$$

Yn awr mae'r canlyniadau hyn yn ffurfio trydedd res y tabl:

7	1
5	35
2	6

Gall y rhes gyntaf y tabl unwaith eto yn cael ei diddymu:

5	35
2	6

Mae angen i ni wneud y dilyniant o gamau unwaith eto:

Rydym yn rhannu'r gwerth colofn chwith y rhes gyntaf gan werth colofn chwith yr ail res, yn ysgrifennu canlyniad fel rhif cyfan ac yn anwybyddu unrhyw relyw.

5	35
2	6
2	

Rydym yn lluosi'r canlyniad hwn gan bob un o'r gwerthoedd yr ail res:

5	35
2	6
2 * 2 = 4	2 * 6 = 12

Yna mae cyfansymiau hyn yn cael eu tynnu o'r gwerthoedd ar y rhes uchaf:

5	35
2	6
(5 – 4) = 1	(35-12) = 23

Mae'r canlyniadau hyn yn awr yn ffurfio trydedd res y tabl:

7	1
5	35
1	23

Mae'r golofn ar ochr chwith y rhes olaf bellach yn cynnwys gwerth o 1, sy'n golygu bod ateb wedi'i ganfod. Mae gwerth ei angen ar gyfer y cod dadgriptio **d** yw'r gwerth cyfatebol yn y golofn dde. Felly, rydyn ni'n cael:

$$d = 23$$

Mae pob un o'r gwerthoedd sydd eu hangen i weithredu'r system amgryptio allweddi dwbl yn awr ar gael i'w defnyddio:

$$n = 55,\ e = 7,\ d = 23$$

n ac **e** gyda'i gilydd yn ffurfio'r allwedd gyhoeddus, tra **n** a **d** gyda'i gilydd yn ffurfio'r allwedd breifat.

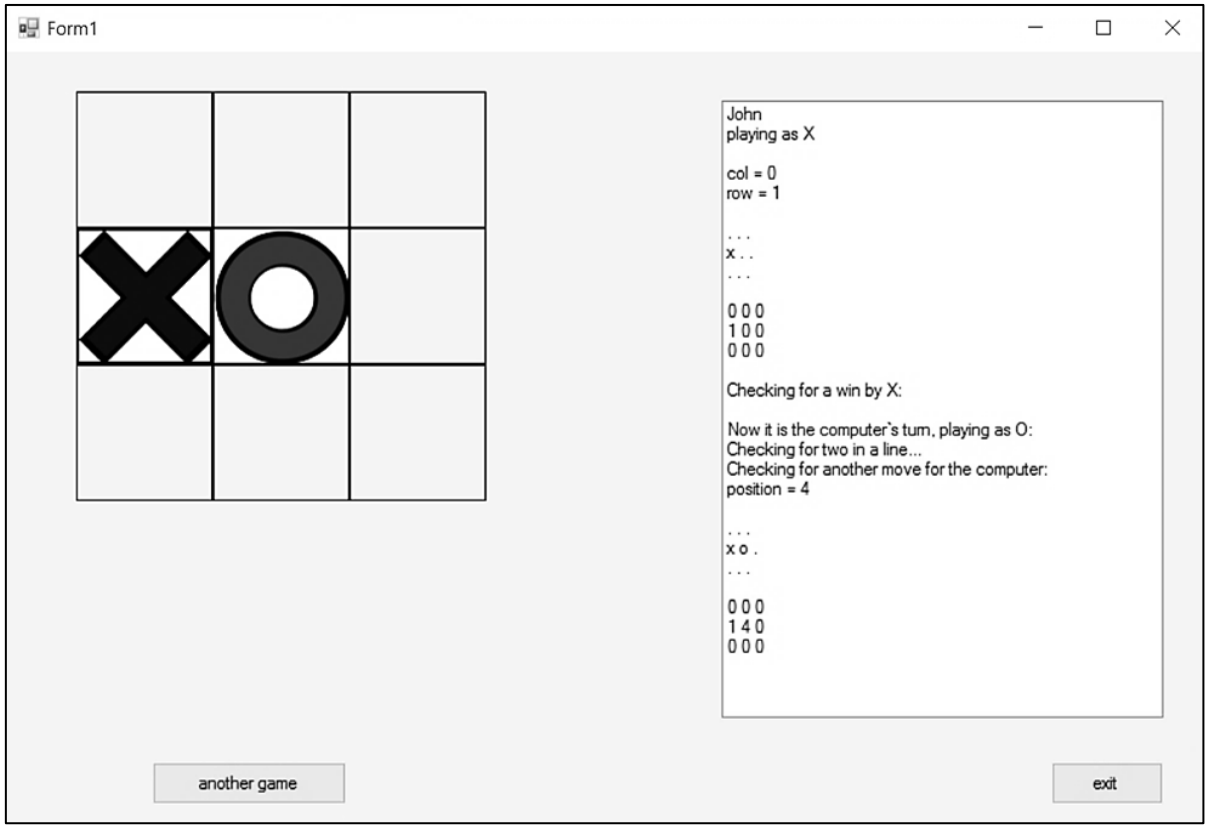
Strategaeth gêm

Mae algorithmau yn bwysig mewn strategaethau cynllunio ar gyfer llwyddiant mewn gemau. Ar wahân i rôl mewn gemau a ddatblygwyd ar gyfer hamdden, gall algorithmau fod yn bwysig wrth fodelu senarios byd go iawn cymhleth megis argyfyngau economaidd neu wrthdaro milwrol. Ar gyfer llawer o'r cymwysiadau hyn, mae adnabod patrymau ac amcangyfrif tebygolrwydd yn bwysig.

Gêm Cylchoedd a Chroesau

Rydym yn dechrau gyda'r gêm syml a chyfarwydd o Gylchoedd a Chroesau. Mae hyn yn cael ei chwarae ar rid o dair rhes o dri sgwâr. Dau chwaraewr, a ddynodwyd yn gylch 'O' neu groes 'X', yn cymryd eu tro i ychwanegu eu symbolau i'r bwrdd. Amcan y gêm yw sicrhau llinell o dri chylch neu groes, naill ai yn llorweddol, fertigol neu groeslinol.

Fel her ddiddorol, gall myfyrwyr cyfrifiadureg cynhyrchu fersiwn o'r gêm lle mae un chwaraewr yn cystadlu yn erbyn y cyfrifiadur. Er mwyn cyflawni hyn, mae'n rhaid i'r cyfrifiadur gael ei raglennu gyda strategaeth gêm.



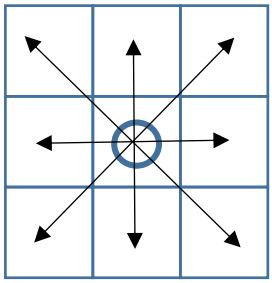
Ffigur 448: Gêm cylchoedd a chroesau, gan gynnwys sylwadau i esbonio'r strategaeth cyfrifiadur

Mae'r chwaraewr yn cael ei ddynodi fel X, a'r cyfrifiadur fel O.

Ar ôl pob symudiad gan y chwaraewr, mae'r cyfrifiadur yn cymryd rheolaeth ac yn gallu gwneud symudiad. Mae'n rhaid iddo, fodd bynnag, gwiriwch yn gyntaf a yw'r chwaraewr newydd gwblhau llinell o dri X ac wedi ennill y gêm.

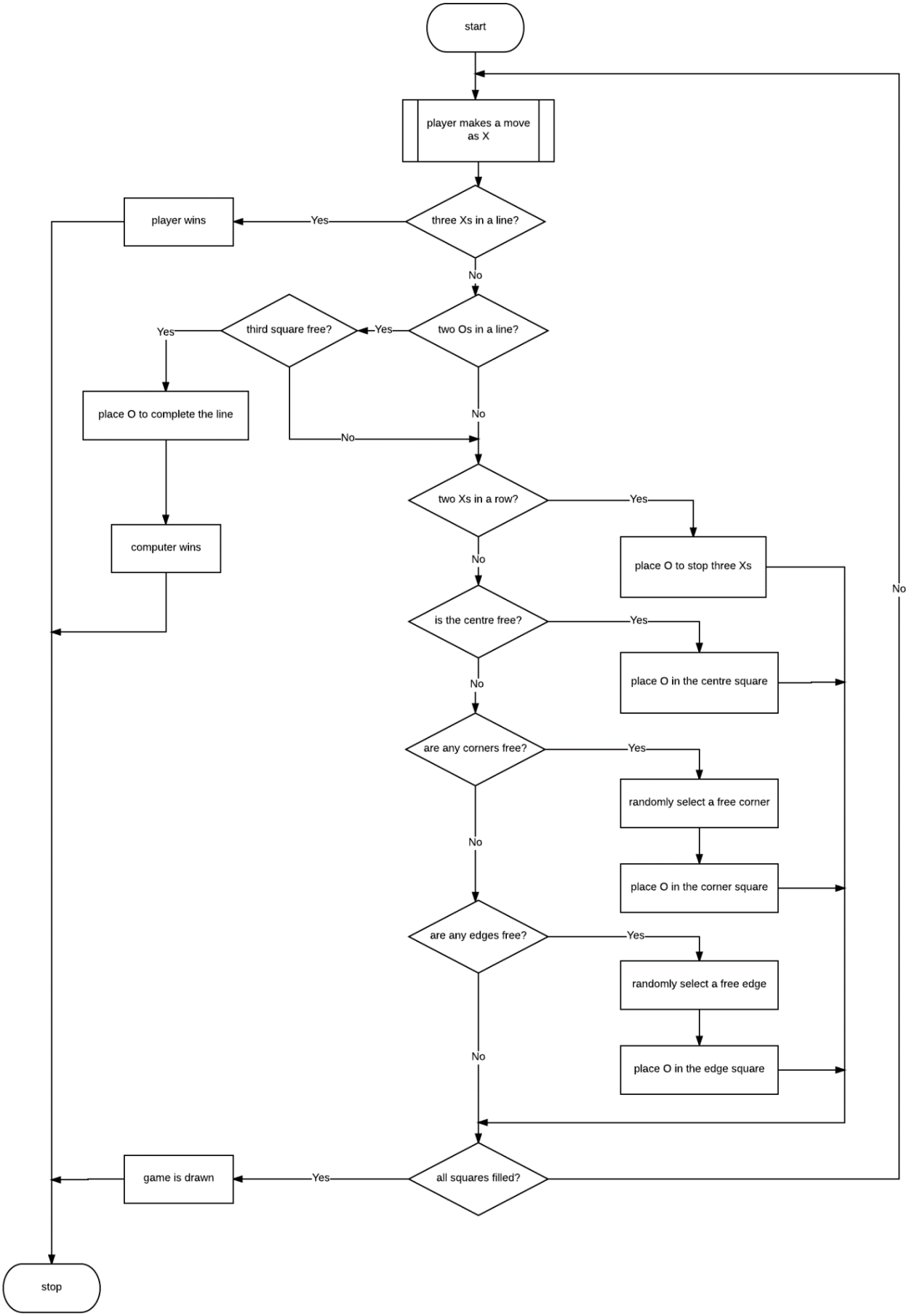
Os bydd y gêm yn parhau, mae'r cyfrifiadur gwiriadau nesaf a oes llinell gyda dau O a lle gwag. Os felly, gall y cyfrifiadur gwblhau'r O derfynol ac ennill y gêm.

Os bydd y gêm yn dal i barhau, mae'n rhaid i'r cyfrifiadur ceisio gwneud symud. Gallwn weld nad oedd yr holl sgwariau ar y bwrdd yr un mor bwysig o ran strategaeth. Gall y sgwâr canolog dod yn rhan o nifer fwyaf o linellau buddugol:



Ffigur 449:

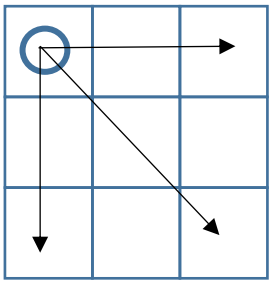
Llinellau buddugol sy'n mynd drwy'r sgwâr canolog



Ffigur 450: Siart llif ar gyfer y gêm cylchoedd a chroesau cyfrifiadurol

Mae mantais i'r cyfrifiadur i ddewis y sgwâr canolog os yw hyn yn dal i fod ar gael.

Os bydd y sgwâr canolog eisoes yn cael ei feddiannu, yna bydd y strategaeth orau nesaf yw dewis sgwâr cornel os yw un ar gael.

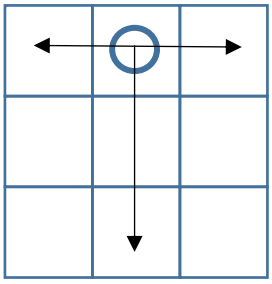


Ffigur 451:

Llinellau buddugol yn pasio drwy sgwâr cornel

Os bydd mwy nag un sgwâr cornel yn wag, mae'r cyfrifiadur yn dewis cornel ar hap. Mae hyn yn rhoi elfen o siawns ac yn caniatáu rhywfaint o gyfle i'r chwaraewr i guro'r cyfrifiadur.

Os bydd y sgwariau cornel i gyd wedi cael eu llenwi eisoes, y dewis olaf bydd sgwâr ymyl os yw un ar gael.



Ffigur 452:

Llinellau buddugol yn pasio drwy sgwâr ymyl

Os bydd mwy nag un sgwâr ymyl yn wag, mae'r cyfrifiadur eto yn dewis ar hap.

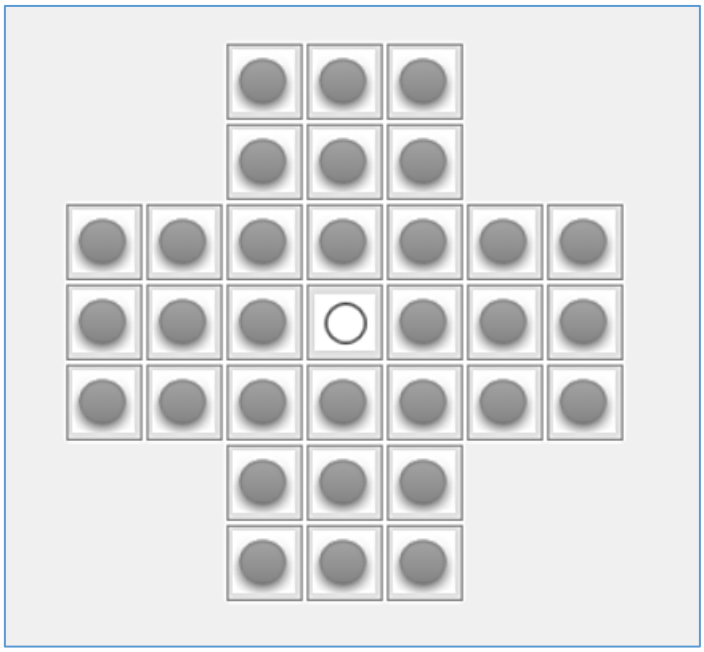
Unwaith bydd y cyfrifiadur wedi gwneud ei thro, mae rheolaeth yn dychwelyd i'r chwaraewr. Fodd bynnag, os nad oes unrhyw sgwariau dal yn rhydd ar y bwrdd, yna bydd y gêm wedi dod i ben yn gyfartal.

Solitaire

Ein henghraifft nesaf yw'r gêm o solitaire. Mae hyn yn cael ei chwarae ar fwrdd sy'n cynnwys patrwm siâp-croes o dyllau y gellir mewnosod pegiau. Ar ddechrau'r gêm, yr holl dyllau, ac eithrio'r twll canolog, yn cynnwys pegiau (ffigur 453).

Mae solitaire yn cael ei chwarae gan chwaraewr sengl. Yr amcan yw cael gwared ar bob un ond un o'r pegiau oddi ar y bwrdd, gan adael y peg olaf yn y twll canolog. Gall peg cael ei diddymu gan neidio peg arall drosto i gyrraedd twll gwag. Gall symud ddigwydd yn llorweddol neu'n fertigol, ond nid ar letraws.

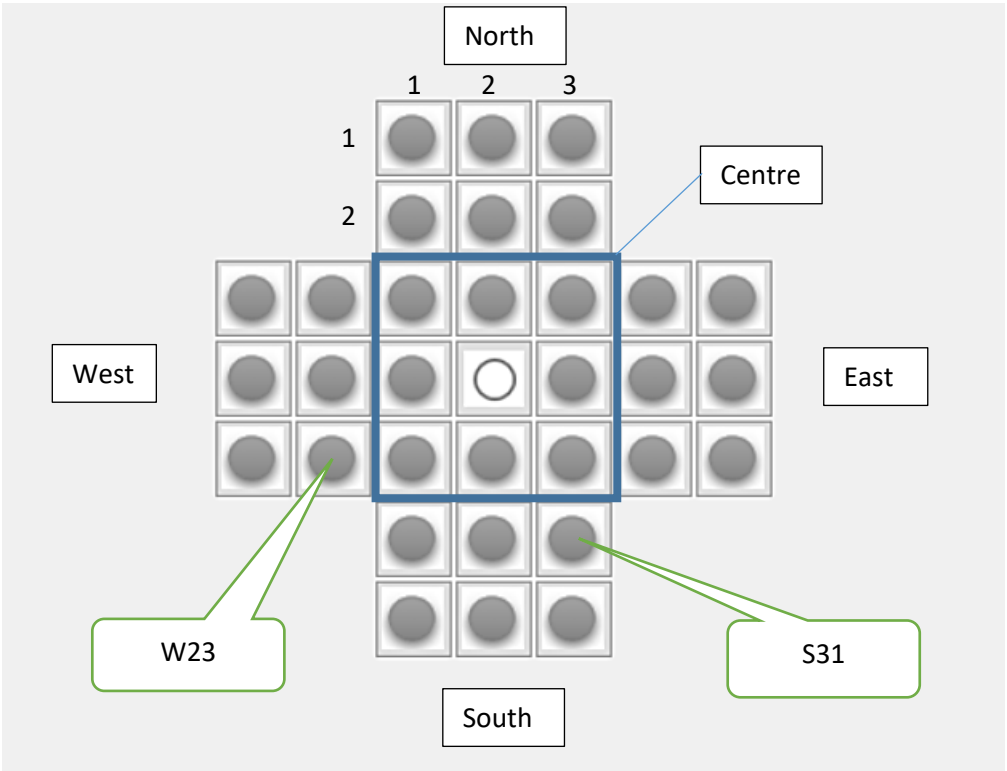
Yn yr adran hon, rydym yn rhoi algorithm ar gyfer cwblhau'r pos solitaire. Gall myfyrwyr cyfrifiadureg un cael her o gynhyrchu dilyniant ateb awtomataidd. Gellir animeiddiad yn cael ei greu gan ddefnyddio graffeg a dynnwyd gan y rhaglen.



Ffigur 453:

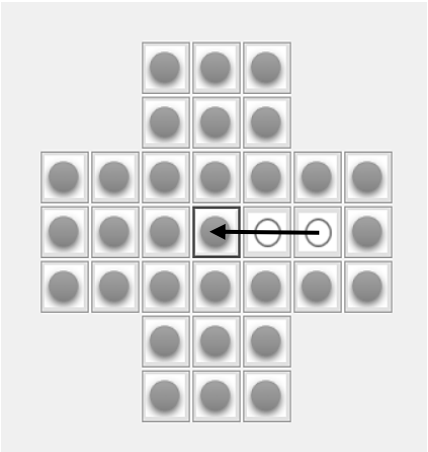
Bwrdd solitaire ar ddechrau'r gêm

Wrth ysgrifennu algorithm y gêm, bydd angen system ar gyfer pennu'r symudiadau. Rydym yn dechrau drwy greu system gydlynu i'r tyllau. Mae'r bwrdd wedi ei rhannu'n bum ardal o'r enw yn Saesneg: North, South, East, West a Centre, fel y dangosir yn ffigur 454. O fewn pob un o'r meysydd hyn, lleoliadau twll unigol yn cael eu nodi gan gyfesuryn llorweddol, a ddilynir gan gyfesuryn fertigol.

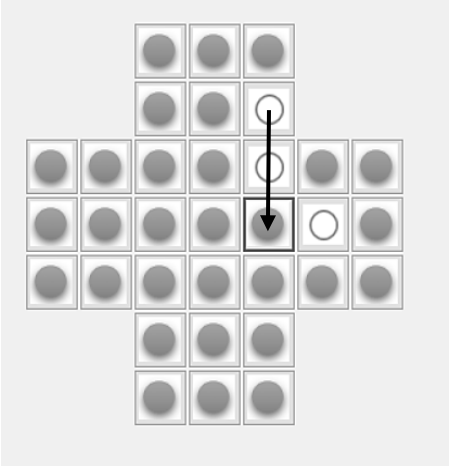


Ffigur 454: System cyfesurynnau ar gyfer y bwrdd solitaire

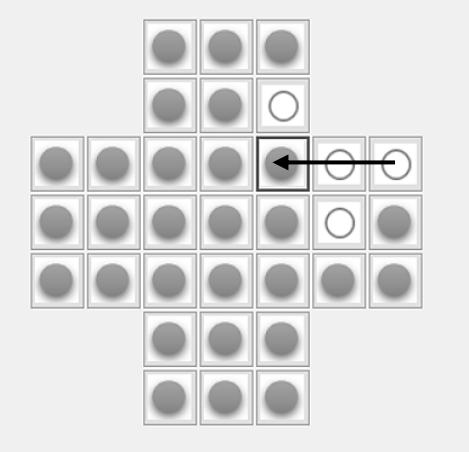
Yr amcan cyntaf yw clirio'r rhan fwyaf o'r pegiau o'r rhan ddwyreiniol y bwrdd. Cyflawnir hyn mewn cyfres o symudiadau:



E12 – C22

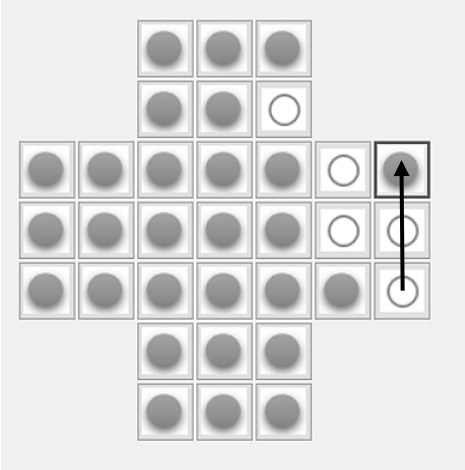


N32 – C32

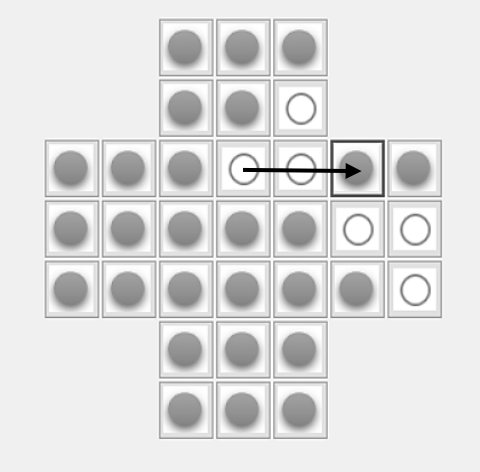


E21 – C31

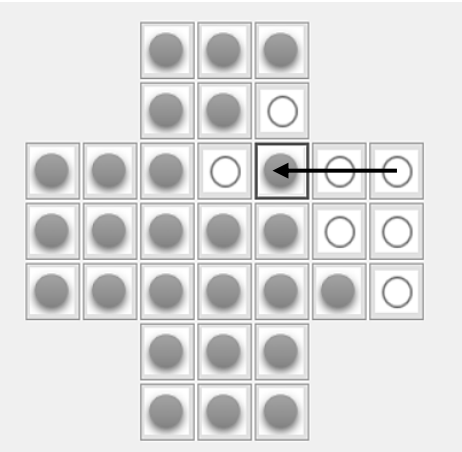
Ffigur 455: Dilyniant o symudiadau i glirio'r rhan Ddwyrain y bwrdd



E23 – E21



C21 – E11



E21 – C31

Ffigur 455 (parhad): Dilyniant o symudiadau i glirio'r rhan Ddwyrain y bwrdd

Mae hyn yn cwblhau clirio'r ardal Ddwyrain y bwrdd. Gall ardal y Gogledd bellach yn cael eu clirio mewn ffordd debyg, gan ddefnyddio dilyniant o symudiadau a restrir yn ffigur 456.

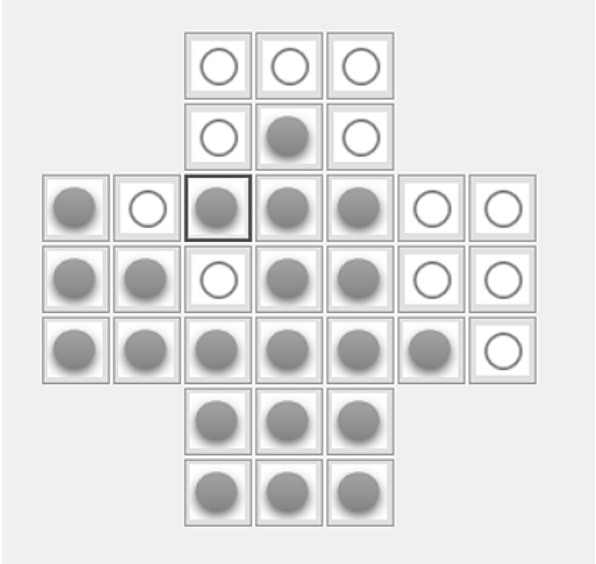
W21-C21

N11-C11

N31-N11

C12-N12

N11-C11



Ffigur 456: Dilyniant o symudiadau i glirio'r rhan Gogledd y bwrdd

Gall ardal y Gorllewin yn cael ei glirio yn yr un modd, gan ddefnyddio dilyniant o symudiadau yn ffigwr 457.

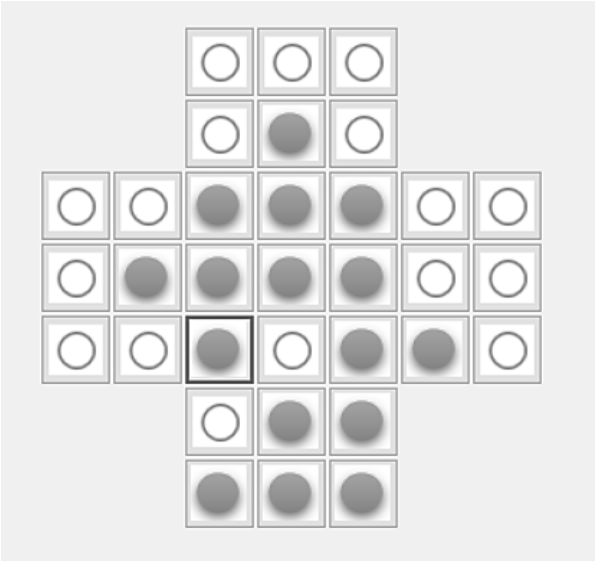
S11-C12

W13-C13

W11-W13

C23-W23

W13-C13



Ffigur 457: Dilyniant o symudiadau i glirio'r rhan Gorllewin y bwrdd

Rydym yn olaf clirio adran De'r bwrdd, gan adael patrwm siâp saeth yng nghanol y bwrdd, fel y dangosir yn ffigwr 458 isod.

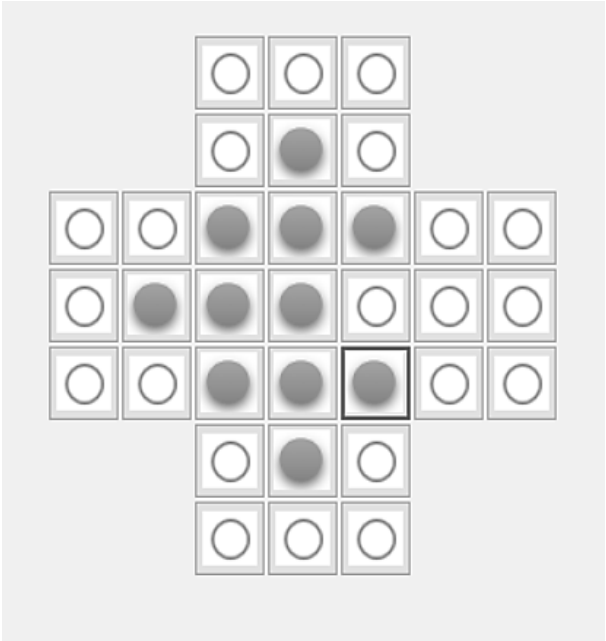
E13-C23

S32-C33

S12-S32

C32-S31

S32-C33



Ffigur 458: Dilyniant o symudiadau i glirio'r rhan De'r bwrdd

Maen bosibl nawr i symud un peg o amgylch y bwrdd mewn cyfres o neidiau, yn cael gwared â phegiau ar ei ffordd ac yn gadael siâp-T syml.

C12-N12

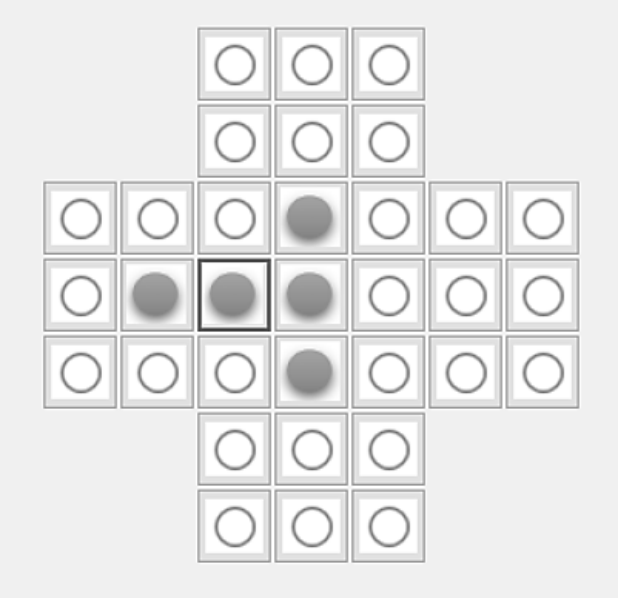
N12-N32

N32-C32

C32-S31

S31-S11

S11-C12



Ffigur 459: Dilyniant o symudiadau i greu'r siâp-T terfynol

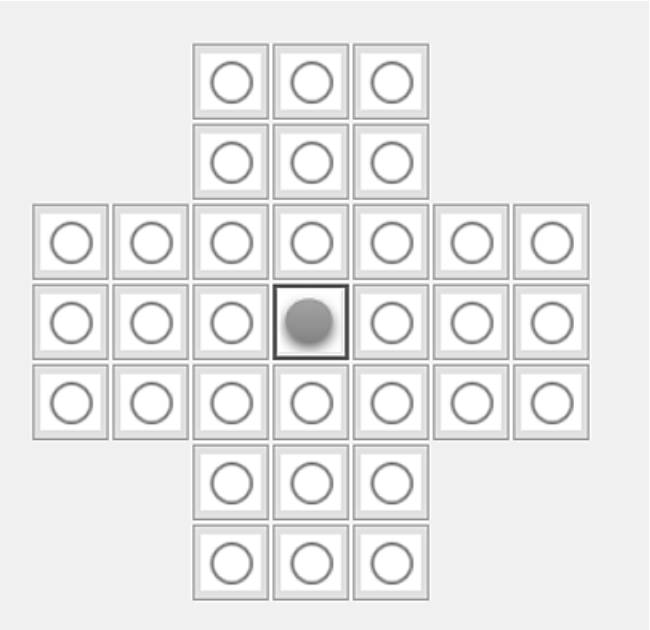
Gall y gêm yn cael ei gwblhau gyda dilyniant terfynol o symudiadau a ddangosir yn ffigur 460.

C22-N22

W22-C22

C23-C21

N22-C22



Ffigur 460: Set derfynol o symudiadau i gwblhau'r datrysiad

Crynodeb

Mae algorithmau yn cael sawl nodwedd yn gyffredin:

- Yn ystod pob cylch o'r algorithm, efallai bod prawf yn cael ei wneud (ee a geiriau yn y drefn gywir, neu gall gyrraedd tref mewn pellter byrrach), yna gellir rhywfaint o newid ei gwneud i'r data (ee trefn geiriau yn cael ei newid, neu gyfanswm o bell yn cael ei ddiweddaru).
- Efallai y bydd camau'r algorithm yn cael ei ailadrodd nifer o weithiau nes i rai amcan wedi ei gyflawni (ee set o eiriau nawr yn nhrefn yr wyddor gywir, neu'r pellter byrraf ar gyfer taith wedi ei ganfod).

Gall algorithm yn darparu dilyniant o gyfarwyddiadau i arwain rhywun trwy dasg gymhleth (ee ddatrys y pos solitaire) neu ddarparu dilyniant o orchmynion ar gyfer rhaglen gyfrifiadurol i wneud y dasg (ee cynhyrchu setiau o allweddi ar gyfer amgryptio allweddi dwbl).

Gall dylunio algorithmau effeithlon yn gofyn am lefelau uchel o sgiliau rhifedd mewn datrys problemau, adnabod patrymau, ac efallai gwybodaeth o dechnegau mewn cymhwyso rhif, geometreg neu algebra. Yn aml mae algorithmau yn cael eu rhedeg gan raglenni cyfrifiadurol, felly gall dealltwriaeth o systemau technoleg gwybodaeth hefyd yn bwysig.