TÖL403G GREINING REIKNIRITA

21. Netareiknirit 1

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Netafræði (*graph theory*)
  - Saga neta
  - Helstu skilgreiningar
  - Notkunardæmi

5.1 − 5.6

- Reiknirit fyrir net
  - Gagnagrindur fyrir net
  - Skoða alla hnúta nets (*graph traversal*)
    - Djúpleit
    - Breiðleit
    - Bestleit

Netafræði - saga

Geta verið tölur, fólk, borgir, vefsíður, ...

- Net eru safn para af hlutum

- Táknum einstaka hluti sem hnúta (*vertices*, *nodes*)
- Einstök pör eru kölluð stikur (*edges*) eða örvar (*arcs*)

Fyrstu dæmi um notkun neta:

Net þjóðvega í Rómaveldi

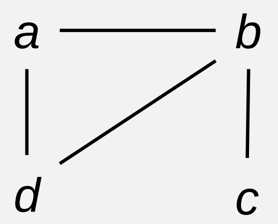
Rómverskir verkfræðingar teiknuðu upp alla vegi sem tengdu saman borgir

Táknun ættfræðiupplýsinga

Notað m.a. af kaþólsku kirkjunni til að ákveða hvort skyldmenni mættu giftast

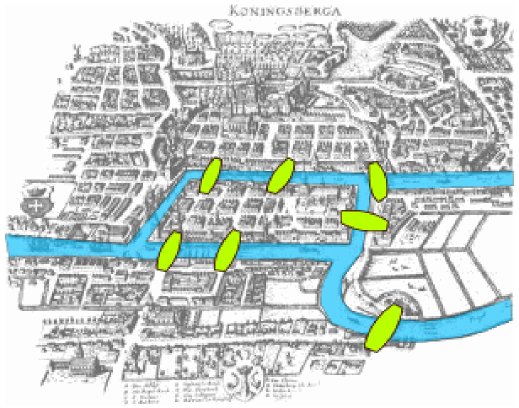
Dæmi:

$$\{ \{a, b\}, \{a, d\}, \{b, c\}, \{b, d\} \}$$



Brýrnar í Köningsberg

- Þekktasta sögulega netafræðiverkefnið



Frá Wikipedia

Borgin Köningsberg liggur beggja vegna árinnar Pregel
Þar eru tvær eyjur: Kneiphof og Lomse

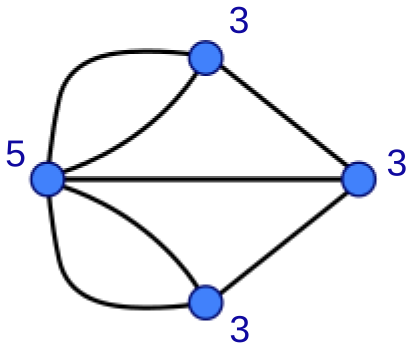
Sjö brýr liggja yfir ánna

Er hægt að ganga yfir allar sjö brýrnar, hverja þeirra nákvæmlega einu sinni og enda á sama stað og maður byrjar?

Stærðfræðingurinn Leonhard Euler leysti verkefnið árið 1735

Hann sannaði að slíkur hringur sé aðeins mögulegur í neti
þar sem allir hnútarnir hafa jafntölu gráður

Í netinu sem brýrnar mynda eru allar gráðurnar oddatala!



Ýmsar skilgreiningar

Net (*graph*) G er tvennd af mengjum (V, E), þar sem

V er mengi hnúta (*vertices*) og

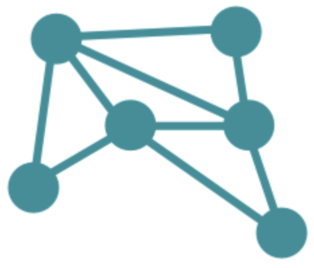
E er mengi hnútapara, sem kallast stikur (*edges*)

Í óstefndu neti eru stikurnar tveggja staka mengi $\{u, v\}$, oft skrifað ***uv***

Í stefnuneti eru stikurnar raðað hnútapar (u, v), oft skrifað $\boldsymbol{u \rightarrow v}$

Einföld net (*simple graphs*) leyfa hvorki *lykkjur* (þ.e. stikuna (u, u)) né samhliða stikur

Athugið að E er mengi!



Net sem leyfa lykkjur og samhliða stikur kallast fjölnet (*multi graphs*)

Við notum sjaldan fjölnet

Netið úr brúunum í Köningsberg var fjölnet

Líka til ofurnet (*hyper graph*), þar sem stikur geta tengt saman fleiri en tvo hnúta

Fleiri skilgreiningar

- Granni (*neighbor*) hnútar v
  - Allir hnútar u sem eru tengdir v með stiku
- Gráða (*degree*) hnútar v
  - Fjöldi nágranna v
- Hlutnet (*subgraph*) $G = (V, E)$
  - Annað net $G' = (V', E')$, með $V' \subseteq V$ og $E' \subseteq E$
- Vegur (*walk*) í neti G
  - Runa hnúta þar sem samliggjandi hnútar eru grannar í G
- Einfaldur vegur (*path*) í neti G
  - Vegur þar sem enginn hnútur er endurtekinn
- Hringur (*cycle*) í neti G
  - Vegur sem byrjar og endar í sama hnúti
- Óhringað (*acyclic*) net
  - Net sem hefur enga hringi

Samhengisþáttur (*connected component*) er hlutnet þar sem það er vegur á milli allra hnúta

Tré (*tree*) er samhangandi óhringað net

Spantré (*spanning tree*) nets G er tré með öllum hnútum netsins G

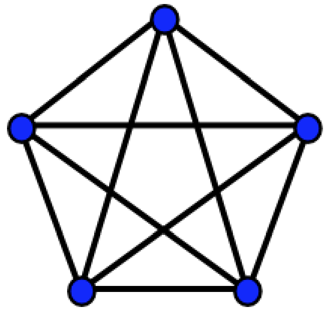
Framsetning neta

- Algengt að sýna net með því að teikna þau í sléttunni (*plane*)
  - Hnútar teiknaðir sem hringir og stikur sem línur eða ferlar á milli þeirra
- Net er <u>sléttunet</u> (*planar graph*) ef hægt er að teikna það í sléttunni án þess að nokkrar línu skerist
  - Slík teikning kallast <u>greyping</u> (*embedding*)
- Það er oft erfitt að sjá hvort net sé sléttunet

Þau mega ekki hafa of margar stikur!

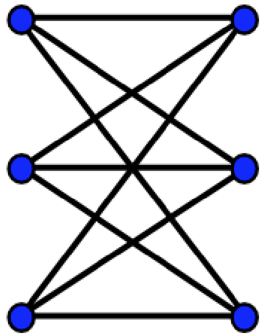
Minnstu net sem ekki eru sléttunet eru K_5 og $K_{3,3}$

Hægt að sýna að net *G* er sléttunet ef það felur ekki í sér K_5 eða $K_{3,3}$



K_5

Fullkomna netið á 5 hnútum



$K_{3,3}$

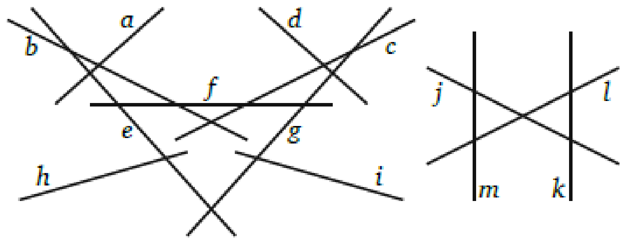
Fullkomna tvíflokka netið á 3 og 3 hnútum

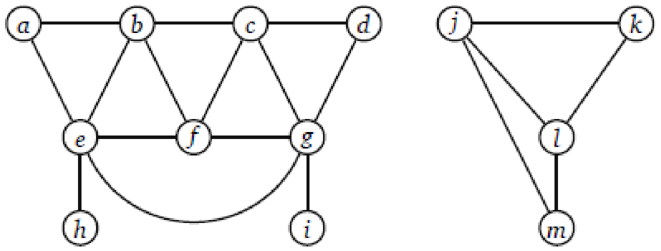
Aðrar framsetningar neta

- <u>Skurðnet</u> (*intersection graph*) fyrir hluti af ýmsu tagi

  Hnútur fyrir hvern hlut   Stika fyrir hverja tvo hluti sem skarast

  <u>Dæmi</u>:





Hver lína er hnútur

Stika ef línur skerast

- <u>Tenginganet</u> (*dependency graph*)

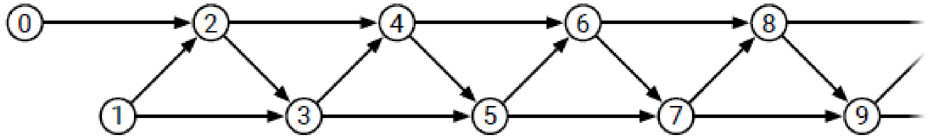
  Hnútur fyrir hvert undirverkefni

  Ör ef lausn eins undirverkefnis byggir á lausn annars

<u>Dæmi</u>:

$$F_n = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

Hnútur i er undirverkefni F_i



Fleiri framsetningar

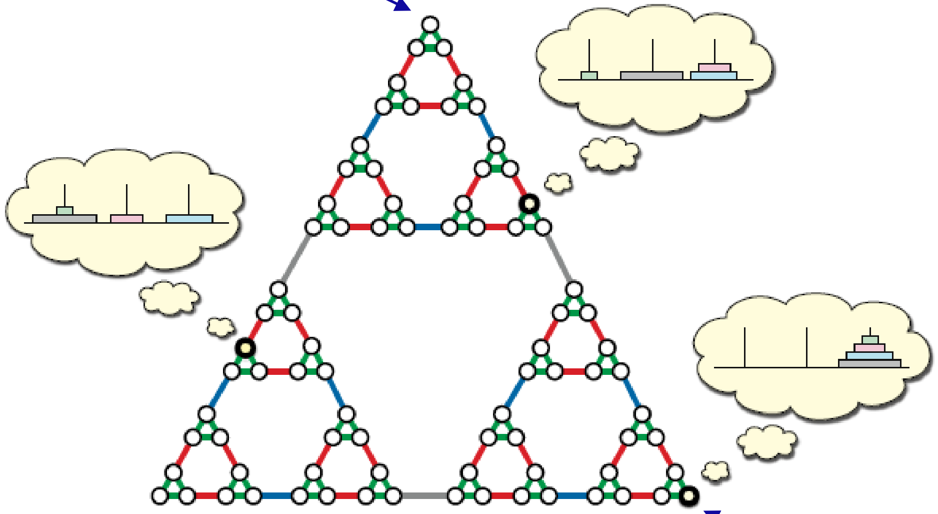
- Stöðunet (*configuration graph*) leikja

  Hver hnútur í netinu er lögleg staða í leiknum

  Stika á milli hnúta ef hægt er að koma á milli staðanna með löglegum leik í stöðunni

Dæmi: Turnarnir í Hanoi 3 súlur, *n* skífur

Upphafsstaða

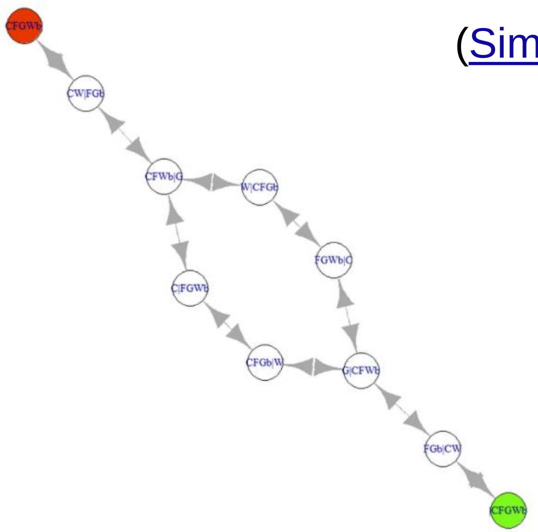


sjá líka Wikipedia

Lokastaða

Annað dæmi: Úlfur, lamb og heypoki

(Simpsons útgáfan)

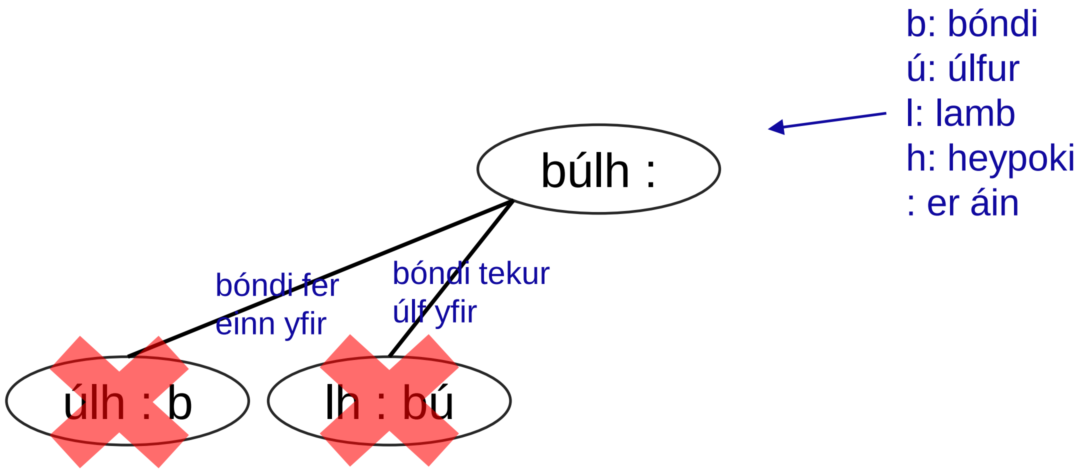


Leikjatré (game trees) eru nátengd stöðunetum

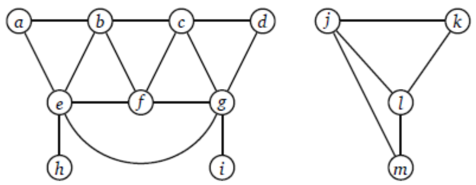
Þar má sama staða þó koma upp oftar en einu sinni

Æfingar

- Teiknið upp fyrstu 3 löglegu hnútana í stöðunetinu fyrir Úlfur-lamb-heypoki

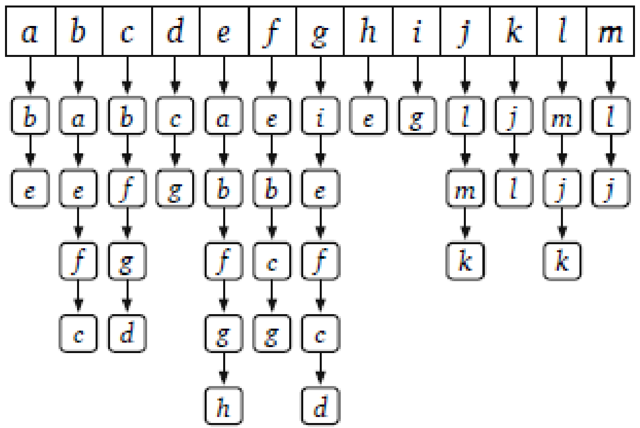


Gagnagrindur neta



- <u>Grannlisti</u> (*adjacency list*)
  - Einvítt fylki af tengdum listum
  - Hver listi inniheldur nágranna eins hnútar
  - Minnisnotkun er $O(V + E)$

Notum nöfn mengjanna sem stærðir þeirra!



- Góð gagnagreind ef farið er í gegnum allt netið
  - Notar lítið minni
  - Dýrt að finna hvort tiltekin stika sé til staðar

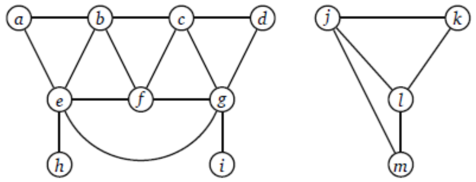
Til ýmsar útgáfur af grannlistum:

- Nota tvíleitartré í stað tengdra lista
- Nota hakkatöflu í stað tengdra lista
- Geyma allar stikurnar í einu fylki og vísa í það

a	b	c	d	e	f	g	h	i	j	k	l	m
1	3	7	11	13	18	22	27	28	29	32	34	37

2	5	1	3	5	6	2	4	6	7	3	7	1	2	6	7	8	2	3	5	7	3	4	5	6	9	5	7	11	12	13	10	12	10	11	13	10	12

Gagnagrindur neta



- <u>Grannfylki</u> (*adjacency matrix*)

  V x *V* fylki af 0-um og 1-um

  Yfirleitt geymt í minni sem tvívítt fylki *A*[1..*V*, 1..*V*]

  $$A[u, v] = 1 \text{ þþaa } uv \in E$$

  Fylki fyrir óstefnt net er alltaf samhverft (*symmetric*)

  Hornalínustökin *A*[*u*, *u*] eru alltaf 0

  Notar $O(V^2)$ minnispláss

  Tekur O(1) tíma að ákvarða hvort stika sé til staðar

  Tekur *O*(*V*) tíma að finna alla nágranna hnútar

Óháð því hversu marga nágranna hann hefur!

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
a	0	1	0	0	1	0	0	0	0	0	0	0	0
b	1	0	1	0	1	1	0	0	0	0	0	0	0
c	0	1	0	1	0	1	1	0	0	0	0	0	0
d	0	0	1	0	0	0	1	0	0	0	0	0	0
e	1	1	0	0	0	1	1	1	0	0	0	0	0
f	0	1	1	0	1	0	1	0	0	0	0	0	0
g	0	0	1	1	1	1	0	0	1	0	0	0	0
h	0	0	0	0	1	0	0	0	0	0	0	0	0
i	0	0	0	0	0	0	1	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	0	1	1	1
k	0	0	0	0	0	0	0	0	0	1	0	1	0
l	0	0	0	0	0	0	0	0	0	1	1	0	1
m	0	0	0	0	0	0	0	0	0	1	0	1	0

Samantekt á gagnagrindum

	Standard adjacency list (linked lists)	Fast adjacency list (hash tables)	Adjacency matrix
Space	$\Theta(V+E)$	$\Theta(V+E)$	$\Theta(V^2)$
Test if $uv \in E$	$O(1+\min\{\deg(u),\deg(v)\}) = O(V)$	$O(1)$	$O(1)$
Test if $u \to v \in E$	$O(1+\deg(u)) = O(V)$	$O(1)$	$O(1)$
List v's (out-)neighbors	$\Theta(1+\deg(v)) = O(V)$	$\Theta(1+\deg(v)) = O(V)$	$\Theta(V)$
List all edges	$\Theta(V+E)$	$\Theta(V+E)$	$\Theta(V^2)$
Insert edge uv	$O(1)$	$O(1)^*$	$O(1)$
Delete edge uv	$O(\deg(u)+\deg(v)) = O(V)$	$O(1)^*$	$O(1)$

Stjarna * þýðir væntur jafnaðartími

Virðist oftast hagkvæmast að nota grannlista með hakkatöflum

- Grannfylki er reyndar mjög hagkvæm þegar netið er þétt
- Venjulegir grannlistar eru líka oft nógu góðir fyrir mörg verkefni

Lítið umstang (*overhead*)
við þessar einfaldari aðferðir

Við munum gera ráð fyrir að net séu geymd með venjulegum grannlistum

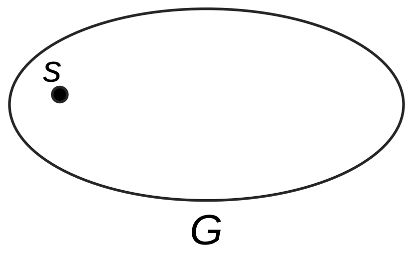
Skoða alla hnúta nets (*traversal*)

- Grundvallarspurning um net:

Sækjanleiki(?) (*reachability*)

Gefið net *G* og hnútur *s* í *G*, í hvaða hnúta er hægt að komast frá *s*?

eða, fyrir hvaða hnút *v* er til vegur frá *s* til *v* í *G*?



Augljósasta aðferðin er að nota <u>djúpleit</u> (*depth-first search*)

```
RecursiveDFS(v):
  if v is unmarked
      mark v
      for each edge vw
          RecursiveDFS(w)
```

Endurkvæm útgáfa

```
IterativeDFS(s):
  Push(s)
  while the stack is not empty
      v ← Pop
      if v is unmarked
          mark v
          for each edge vw
              Push(w)
```

Ítrunarútgáfa

Hlaðinn er "sýnilegur"
í ítrunarútgáfu

Almenn leit

- Getum skilgreint almenna útgáfu af netaleit:

```
WhateverFirstSearch(s):
  put s into the bag
  while the bag is not empty
    take v from the bag
    if v is unmarked
      mark v
      for each edge vw
        put w into the bag
```

Höfum einhverja gagnagrind, poka (*bag*), til að geyma hnútana sem á eftir að skoða

Með því að geyma í pokanum hnútapör, þá getum við smíðað tré fyrir leitina:

```
WhateverFirstSearch(s):
  put (∅, s) in bag
  while the bag is not empty
    take (p, v) from the bag             (⋆)
    if v is unmarked
      mark v
      parent(v) ← p
      for each edge vw                   (†)
        put (v, w) into the bag          (⋆⋆)
```

T er tími fyrir aðgerð á poka

Tími: $O(V + ET)$

Miðað við grannlista, ef grannfylki þá $O(V^2 + ET)$

Öll pörin (v, $parent(v)$) skilgreina spantré samhengisþáttarins sem inniheldur s

Ef v er í sama samhengisþætti og s, þá er hægt að rekja sig aftur til s með:

$v \to parent(v) \to parent(parent(v)) \to \ldots$

Mismunandi tegundir af pokum

- Hlaði (*stack*)

  Fáum þá djúpleit (*depth-first search*)

  Aðgerðirnar á gagnagrindina, *push* og *pop* taka $O(1)$ tíma

  Tími reikniritsins er því $O(V + E)$

  Ef netið er samhangandi, þá er $E > V$, svo þá er tíminn $O(E)$

  Spantréð er þá djúpleitartré (*DFS tree*)

  Lögun þess fer eftir því hvar við byrjum (þ.e. hnúturinn s)

  og eftir heimsóknarröð nágrannanna

  Höldum áfram niður tréð þar til við finnum enga ómerkta hnúta

  Almennt eru djúpleitartré frekar há og grönn

Skoðum djúpleit betur í næsta tíma

Mismunandi tegundir af pokum

- <u>Biðröð</u> (*queue*)

  Fáum þá <u>breiðleit</u> (*breadth-first search*)

  Aðgerðirnar á gagnagrindina, *push* og *pull* taka $O(1)$ tíma

  Tími reikniritsins er því $O(V + E)$

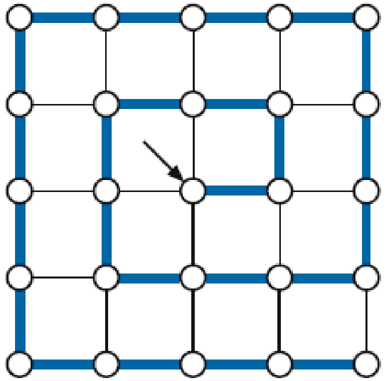
  Ef netið er samhangandi, þá er $E > V$, svo þá er tíminn $O(E)$

  Spantréð er þá <u>breiðleitartré</u> (*BFS tree*)

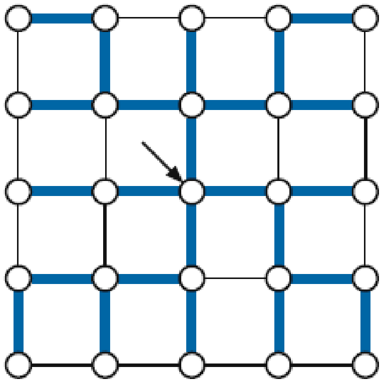
  Lögun þess fer eftir því hvar við byrjum (þ.e. s)

  Skoðum alla nágranna hnútar áður en við förum neðar

  Almennt eru breiðleitartré frekar grunn og breið



Djúpleit, byrjað í miðju



Breiðleit, byrjað í miðju

Mismunandi tegundir af pokum

- <u>Forgangsbiðröð</u> (*priority queue*)

Fáum þá "<u>bestleit</u>" (*best-first search*)

Þetta er eiginlega fjölskylda af reikniritum:

Skiptir máli hvernig forgangurinn er skilgreindur

Ef þyngd stiku er notuð sem forgangur þá fáum við <u>léttasta spantré</u> (*min. spanning tree*)

Reiknirit Prim

Ef fjarlægð hnútar frá upphafshnút er notað sem forgangur þá finnum við <u>stystu vegi</u> (*shortest paths*) í netinu

Reiknirit Dijkstra

Líka hægt að skilgreina forgang til þess að finna <u>flöskuháls stystu vegi</u> (*bottleneck shortest paths*) í netinu

Finnur þann veg frá s til v sem hefur stærstu minnstu stiku (þ.e. hefur sem stærstan flöskuháls)

Finna samhengisþætti nets

- Reikniritið *WhateverFirstSearch* finnur aðeins hnúta í sama samhengisþætti og upphafshnúturinn s
- Til að komast í alla hnúta þurfum við ytra fall:

```
WFSAll(G):
  for all vertices v
      unmark v
  for all vertices v
      if v is unmarked
          WhateverFirstSearch(v)
```

Með smá aukavinnu er hægt að merkja hvern hnút með númeri þáttarins sem hann er í

Getum svo talið samhengisþætti netsins:

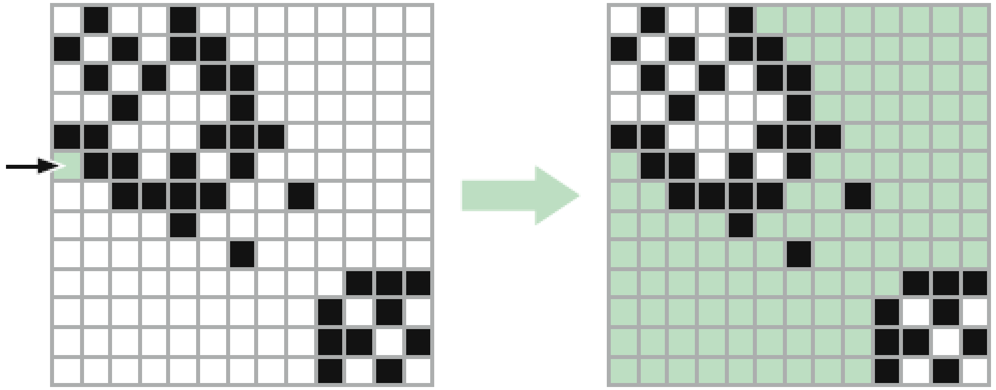
```
CountComponents(G):
  count ← 0
  for all vertices v
      unmark v
  for all vertices v
      if v is unmarked
          count ← count + 1
          WhateverFirstSearch(v)
  return count
```

Sýnidæmi: Flæðisfylling

- Höfum *nxn* skjápunkta (*pixels*)
- Samhangandi svæði (*connected region*) er samhangandi hlutmengi skjápunktanna sem allir hafa sama lit

Tveir skjápunktar eru hliðstæðir ef þeir eru nágrannar lárétt eða lóðrétt

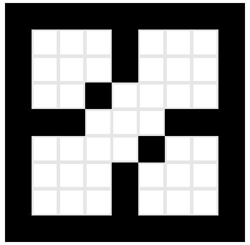
Flæðisfylling (*flood fill*) felst í því að breyta lit allra skjápunkta í samhangandi svæði yfir í nýjan lit



Getum breytt verkefninu yfir í netaverkefni:

Hver skjápunktur verður að hnút í netinu G

Stika á milli aðliggjandi sjápunkta ef þeir hafa sama lit



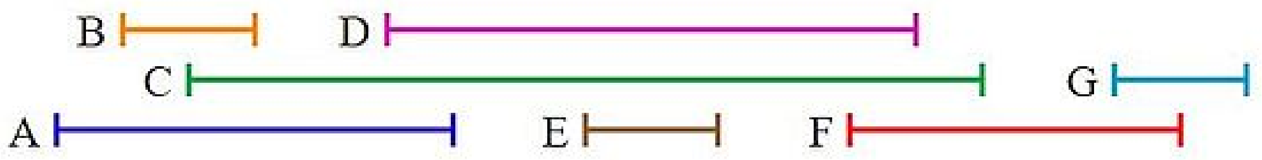
Ef myndin hefur *nxn* skjápunkta, þá hefur netið n^2 hnúta og mest $2n^2$ stikur

Tíminn á almennri leit er $O(V + E)$, sem er $\boldsymbol{O(n^2)}$

Fyrirlestraæfingar

1. Við höfum net með 32 hnútum og 128 stikum.
   a. Ef hægt er að tákna grannfylkið sem bitafylki, hversu mörg bæti tekur þetta net, táknað með grannfylki?
   b. Ef hnútanúmer eru táknuð sem 32-bita heiltölur og bendar eru sömuleiðis 32 bita. Hversu mörg bæti tekur netið ef það er táknað með grannlista?
   c. Hvaða ókostir eru við það að nota bitafylki sem grannfylki í nútímatölvum?

2. Teiknið upp netið sem táknað er með eftirfarandi 7 bilum (*intervals*):



3. Framkvæmið djúpleit (*DFS*) á netið í dæminu að ofan og byrjið í hnúti *A*.