TÖL403G GREINING REIKNIRITA

8. Kvik bestun 2

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Lengsta hækkandi hlutruna (*LIS*)

- *LISbigger* með kvikri bestun
    - Gagnagrind fyrir milliniðurstöður
    - Reiknirit

3.6

- *LISfirst* með kvikri bestun
    - Gagnagrind fyrir milliniðurstöður
    - Reiknirit

Lengsta hækkandi hlutruna (*LIS*)

- Gefið fylki $A[1..n]$ af tölum finna lengd lengstu hækkandi hlutrunu í A
- Sáum síðast tvö endurkvæm endurrakningarreiknirit fyrir verkefnið
- Bæði reiknritin höfðu tímann $O(2^n)$

LISbigger:	Er $A[i]$ næsta stakið í lengstu hækkandi hlutrunu?	Hugsa út frá inntaki
LISfirst:	Hvar er næsta stakið í lengstu hækkandi hlutrunu?	Hugsa út frá niðurstöðu

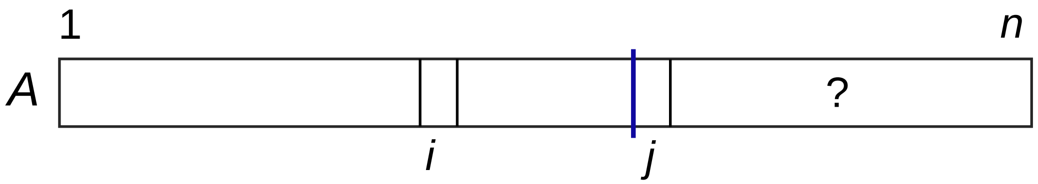
- Byggja á mismunandi endurkvæmum formúlum
- Sjáum nú tvö kvik bestunarreiknirit
  - Byggja á endurkvæmu formúlunum og hafa ólíkar gagnagrindur

Sýnir að hægt er að leysa verkefni á mismunandi vegu með kvikri bestun

LISbigger með kvikri bestun

- Skilgreindum fallið *LISbigger*(*i*, *j*):

LISbigger(*i*, *j*): lengd lengstu hækkandi hlutrunu í *A*[*j*..*n*] með öll stök stærri en *A*[*i*]



<u>Endurkvæm formúla:</u>

$$
LISbigger(i, j) = \begin{cases} 0 & \text{ef } j > n \\ LISbigger(i, j+1) & \text{ef } A[i] \geq A[j] \\ \max\begin{Bmatrix} LISbigger(i, j+1), \\ 1 + LISbigger(j, j+1) \end{Bmatrix} & \text{annars} \end{cases}
$$

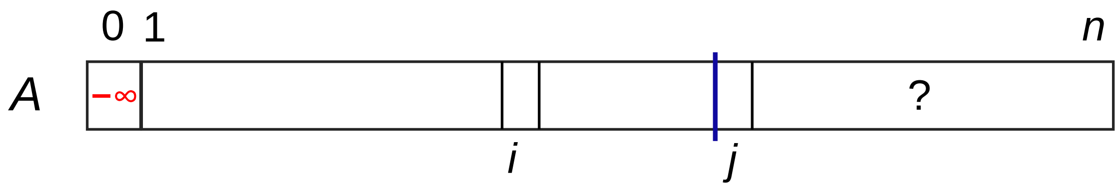
Komin út fyrir fylkið!

Þá kemur *A*[*j*] ekki til greina

Þarf að ákveða hvort taka eigi *A*[*j*] með eða ekki

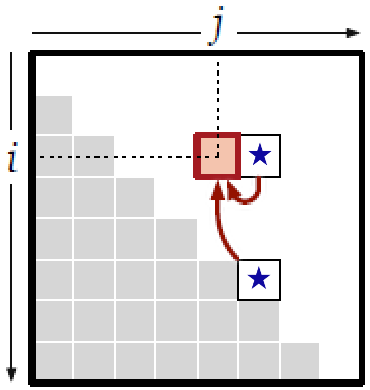
Útgáfa með kvikri bestun

- Bætum við varðstakinu $A[0] = -\infty$



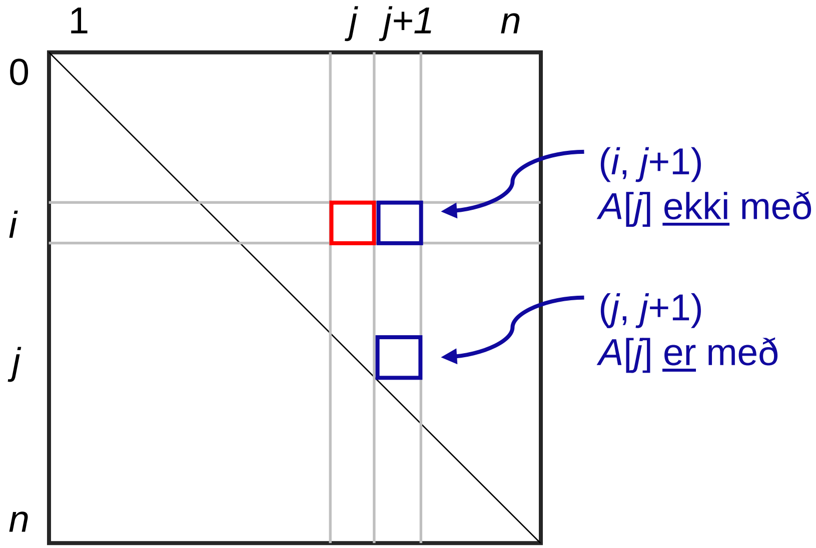
- Finnum svo *LISbigger*(0, 1)
  - þ.e. lengstu hækkandi hlutrunu í $A[1..n]$ með öll stök $> -\infty$

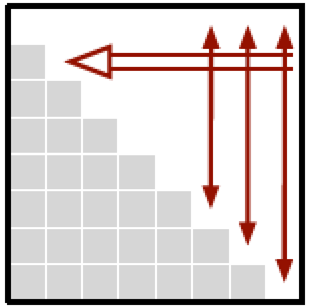
- Skilgreinum tvívíða fylkið *LISbigger*$[0..n, 1..n]$ sem geymir niðurstöður allra hlutverkefnanna
- Til þess að reikna *LISbigger*$[i, j]$ þurfum við:
  - *LISbigger*$[i, j+1]$ ⟵ ef $A[j]$ er <u>ekki með</u> í rununni
  - *LISbigger*$[j, j+1]$ ⟵ ef $A[j]$ <u>er með</u> í rununni



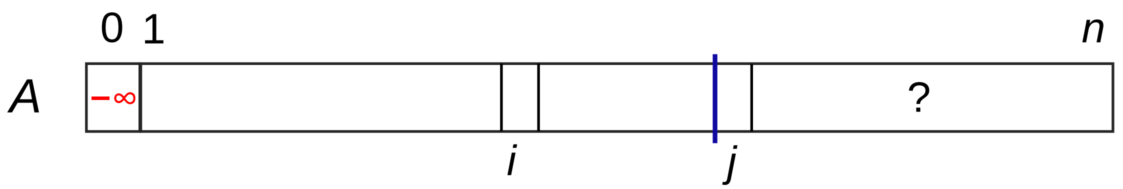
Útreikningsröð

- Fylkið *LISbigger* hefur $O(n^2)$ stök
- Reiknum það dálk fyrir dálk:
  - Stök í dálki j reiknuð út frá dálki j+1
  - Gerum ráð fyrir að dálkur "n+1" sé allur 0
- Við reiknum aðeins gildi fyrir ofan hornalínu
  - Í stökum fyrir neðan er $i > j$, en það getur ekki gengið
  - Stakið $A[i]$ er alltaf vinstra megin við $A[j]$
- Reiknum allan dálk j+1 áður en við byrjum á dálki j
  - Skiptir ekki máli í hvaða röð stök dálkanna eru reiknuð
  - Annað hvort frá i=0 til j−1 (niður) eða frá i=j−1 til 0 (upp)
- Hægt að reikna hvert stak á $O(1)$ tíma





ReiknIritið *FastLIS*




```
FastLIS(A[1..n]):
  A[0] ← −∞                          ⟨⟨Add a sentinel⟩⟩
  for i ← 0 to n                     ⟨⟨Base cases⟩⟩
      LISbigger[i, n+1] ← 0
  for j ← n down to 1
      for i ← 0 to j−1               ⟨⟨…or whatever⟩⟩
          keep ← 1 + LISbigger[j, j+1]
          skip ← LISbigger[i, j+1]
          if A[i] ≥ A[j]
              LISbigger[i, j] ← skip
          else
              LISbigger[i, j] ← max{keep, skip}
  return LISbigger[0, 1]
```

Setja varðstak fyrir framan fremsta stak í *A*

Núllstilla dálk *n*+1 í *LISbigger*

Lengd ef A[j] <u>er</u> með í rununni

Lengd ef A[j] <u>ekki</u> með í rununni

A[*j*] > *A*[*i*], svo A[*j*] getur verið með, en er það betra?

Tími: $O(n^2)$ Minnispláss: $O(n^2)$
hægt að minnka í $O(n)$

Sýnidæmi

Gefin talnarunan [3, 1, 4, 1, 5], svo $n = 5$

Setjum $A[0]$ sem $-\infty$ (hér -100):

LISbigger[*i*, *j*]: lengd lengstu hækkandi hlutrunu í *A*[*j*..*n*] með öll stök stærri en *A*[*i*]

	0	1	2	3	4	5
A	-100	4	1	5	9	2

Fylkið *LISbigger*[0..5, 1..6]

Upphafsstillum aftasta dálk með 0-um

Reiknum svo dálk fyrir dálk fram töfluna

LISbigger

	1	2	3	4	5	6
0						0
1	-					0
2	-	-				0
3	-	-	-			0
4	-	-	-	-		0
5	-	-	-	-	-	0

Sýnidæmi, frh.

Skoðum *LISbigger*[1,3]:

LIS í *A*[3..5] með öll stök > *A*[1]

[5, 9, 2] 4

Sjáum að *A*[3] > *A*[1] (5 > 4), svo *A*[3] getur verið með í LIS í *A*[3..5] með öll stök > *A*[1]

Setjum sem ***max***{ *LISbigger*[1,4], 1+*LISbigger*[3,4] }

A[3] er ekki með

A[3] er með

LISbigger[1,3] = ***max***{ 1, 1+1 } = **2**

LISbigger[*i*, *j*]: lengd lengstu hækkandi hlutrunu í *A*[*j*..*n*] með öll stök stærri en *A*[*i*]

	0	1	2	3	4	5
A	-100	4	1	5	9	2

LISbigger

	1	2	3	4	5	6
0			2	1	1	0
1	-		**2**	1	0	0
2	-	-		1	1	0
3	-	-	-	1 +1	0	0
4	-	-	-	-	0	0
5	-	-	-	-	-	0

Æfingadæmi

- Reiknið *LISbigger*[2, 3]:

LIS í *A*[____] með öll stök > *A*[_]

Getur fremsta stakið í hlutfylkinu verið með?

LISbigger[*i*, *j*]: lengd lengstu hækkandi hlutrunu í *A*[*j*..*n*] með öll stök stærri en *A*[*i*]

	0	1	2	3	4	5
A	-100	4	1	5	9	2

LISbigger

	1	2	3	4	5	6
0			2	1	1	0
1	-		**2**	1	0	0
2	-	-		1	1	0
3	-	-	-	1	0	0
4	-	-	-	-	0	0
5	-	-	-	-	-	0

Sýnidæmi, frh.

	0	1	2	3	4	5
A	-100	4	1	5	9	2

Lokatafla:

LISbigger

	1	2	3	4	5	6
0	3	3	2	1	1	0
1	-	2	2	1	0	0
2	-	-	2	1	1	0
3	-	-	-	1	0	0
4	-	-	-	-	0	0
5	-	-	-	-	-	0

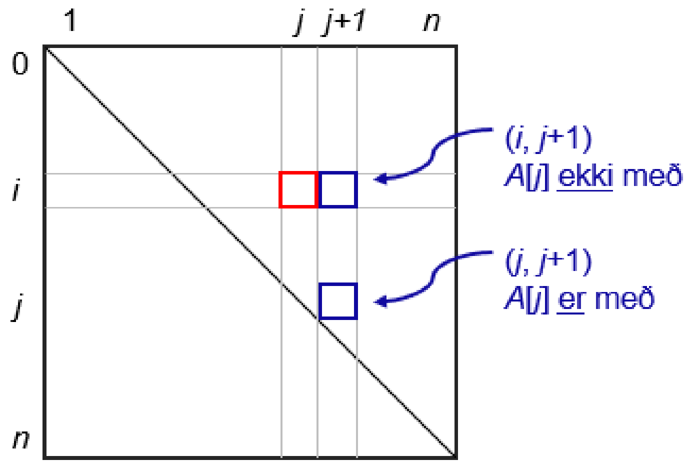
LISbigger[0,1] = 3, svo lengd lengstu hækkandi hlutrunu í *A* er 3

t.d. [4, 5, 9]

Sjáum líka að *LISbigger*[0,2] = 3, svo lengd lengstu hækkandi hlutrunu í *A*[2..5] er líka 3

það er [1, 5, 9]

Að finna hlutrununa



- Lokagildin í *LISbigger* gefa okkur aðeins <u>lengd</u> lengstu hækkandi hlutrunu, en ekki hlutrununa sjálfa
- Til að finna hana þurfum við að geyma upplýsingar um hverja ákvörðun sem er tekin
  - Við útreikning á *LISbigger*[0,1] er fundið ***max***{ *LISbigger*[0,2], 1+*LISbigger*[1,2] }
  - Ef *LISbigger*[0,2] var valið þá er A[1] <u>ekki</u> í lengstu hlutrununni
  - Ef 1+*LISbigger*[1,2] var valið þá <u>er</u> A[1] í lengstu hlutrununni
- Getum geymt upplýsingar um hvort var valið í öðru *nxn* fylki og rakið okkur svo til baka til að finna hvað stök eru í lengstu hlutrununni
  - Tími: $O(n)$

Hin útgáfan: *LISfirst*

- *LISfirst*(*i*) er lengd lengstu hækkandi hlutrunu (*LIS*) í *A*[*i*..*n*] sem byrjar á *A*[*i*]

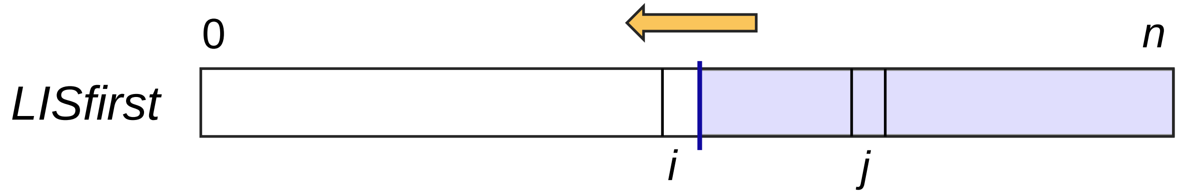
  Endurkvæm skilgreining:

$$LISfirst(i) = 1 + \max\{LISfirst(j) \mid j > i \text{ og } A[j] > A[i]\}$$

- Bætum varðstaki framan við fremsta stakið í *A*
  - Setjum $A[0] = -\infty$
  - Útkoman er þá *LISfirst*(0)−1, því *A*[0] er talið með

- Hér hafa undirverkefnin aðeins vísinn *i*, svo okkur nægir einvíða fylkið *LISfirst*[0..*n*] til að geyma milliniðurstöður
  - Hvert stak *LISfirst*[*i*] byggir aðeins á stökum *LISfirst*[*j*], með $j > i$
  - Getum því reiknað út fylkið aftan frá

Reiknritið *FastLIS2*




```
FastLIS2(A[1..n]):
  A[0] = −∞                          ⟨⟨Add a sentinel⟩⟩
  for i ← n downto 0
      LISfirst[i] ← 1
      for j ← i+1 to n               ⟨⟨... or whatever⟩⟩
          if A[j] > A[i] and 1 + LISfirst[j] > LISfirst[i]
              LISfirst[i] ← 1 + LISfirst[j]
  return LISfirst[0] − 1             ⟨⟨Don't count the sentinel⟩⟩
```

Byrjum aftast og færum okkur fram fylkið

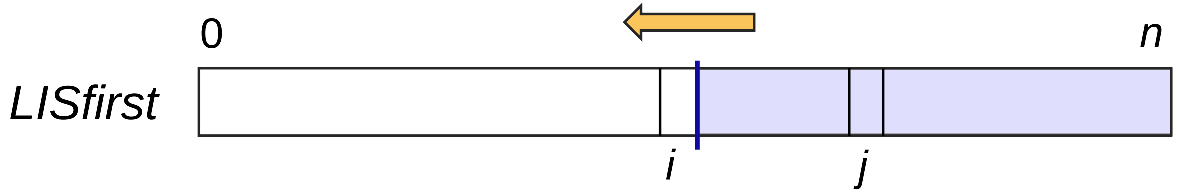
Skoðum svo stökin fyrir aftan *A*[*i*]

Ef *A*[*j*] > *A*[*i*] þá er spurninging: er það betra en það sem við höfum séð hingað til

Sýnidæmi

$$
\begin{array}{ll}
\underline{\text{FastLIS2}(A[1..n]):} & \\
\quad A[0] = -\infty & \langle\langle \textit{Add a sentinel} \rangle\rangle \\
\quad \text{for } i \leftarrow n \text{ downto } 0 & \\
\qquad LISfirst[i] \leftarrow 1 & \\
\qquad \text{for } j \leftarrow i+1 \text{ to } n & \langle\langle \textit{...or whatever} \rangle\rangle \\
\qquad\quad \text{if } A[j] > A[i] \text{ and } 1 + LISfirst[j] > LISfirst[i] & \\
\qquad\qquad LISfirst[i] \leftarrow 1 + LISfirst[j] & \\
\quad \text{return } LISfirst[0] - 1 & \langle\langle \textit{Don't count the sentinel} \rangle\rangle
\end{array}
$$

	0	1	2	3	4	5
A	-100	4	1	5	9	2



	0	1	2	3	4	5
LISfirst				~~1~~ 2	1	1

$i = 5$: *LISfirst*[5] = 1

$i = 4$: *LISfirst*[4] = 1

$j = 5$: $\overset{2}{A[j]} > \overset{9}{A[i]}$ **ósatt**

$i = 3$: *LISfirst*[3] = 1

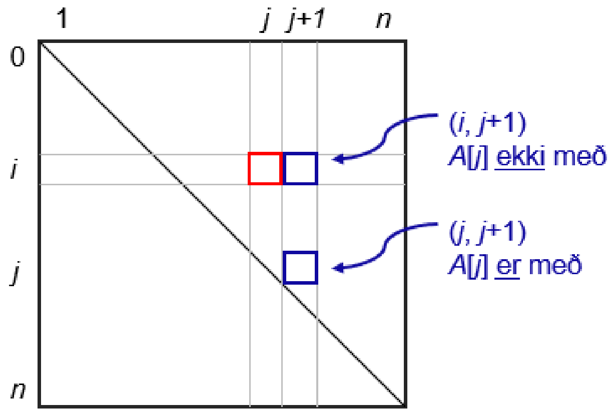
$j = 4$: $\overset{9}{A[j]} > \overset{5}{A[i]}$ **satt** og $\overset{1+1}{1+LISfirst[j]} > \overset{1}{LISfirst[i]}$ **satt** *LISfirst*[3] = 1+*LISfirst*[4]

$j = 5$: $\overset{2}{A[j]} > \overset{5}{A[i]}$ **ósatt**

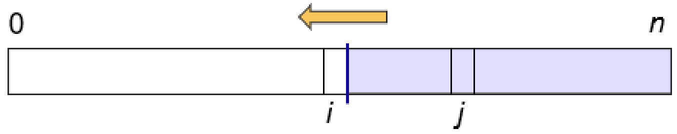
Framhald í heimadæmum!

Samanburður á reiknritum

- *FastLIS* (*LISbetter*) notar $O(n^2)$ minni og $O(n^2)$ tíma
  - Hægt að minnka minnisnotkun í $O(n)$, því við þurfum aðeins síðasta dálk í hvert sinn
  - Reiknar ~$n^2/2$ gildi og hvert þeirra tekur $O(1)$ tíma



- FastLIS2 (*LISfirst*) notar $O(n)$ minni og $O(n^2)$ tíma
  - Reiknar hvert stak i með því að skoða öll stökin fyrir aftan (i+1, ..., n)



- Ekki hægt að segja að önnur aðferðin sé mikið betri en hin
  - Nota ólíka sýn á verkefnið
  - Fáum sömu tímaflækju

Fyrirlestraæfingar

1. Hvaða röð á *n*-staka inntaki gefur lengstu hækkandi hlutrunu? Hvaða röð gefur stystu hækkandi hlutrunu? Hvað giskið þið á að sé **vænt lengd** (*expected length*) á lengstu hækkandi hlutrunu í *n*-staka inntaki í slembiröð (allar *n*! umraðanir jafnlíklegar)?

2. Lát *A* vera [3, 1, 4]. Reiknið út tvívíða fylkið *LISbigger*[0..3, 1..4] samkvæmt kvika bestunarreikniritinu fyrir *LIS*.

3. [Aukaæfing, ef búin með hinar]: Eitt mögulegt gráðugt reiknirit fyrir *LIS* er að finna minnsta stakið í *A* og setja það sem fyrsta stakið í lengstu hækkandi hlutrunu. Finna síðan minnsta stakið sem kemur fyrir aftan það í *A*, setja það sem annað stakið, o.s.frv. Sýnið mótdæmi þar sem þetta reiknirit finnur ekki lengstu hækkandi hlutrunu.