CSCE 750 12/7/2023

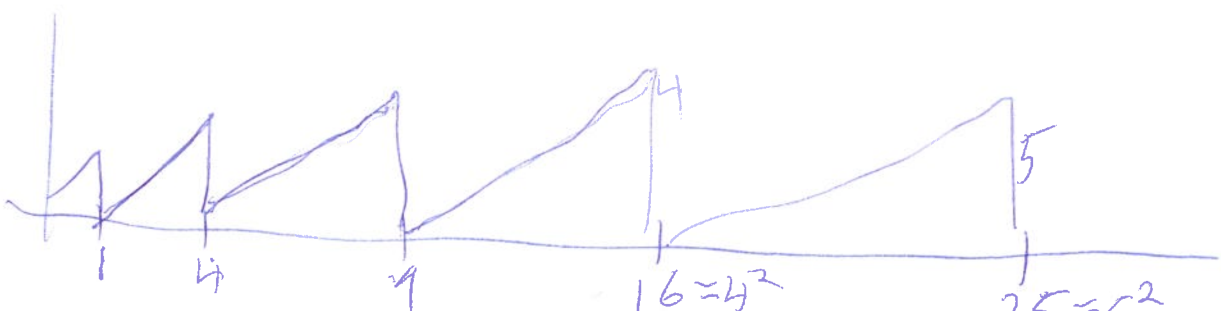
Designing a good potential function

$\Phi \geq 0$ always

$\Phi = 0$ initially

$\hat{c}_i = c_i + \Delta\Phi = O(1)$

$$c_i = \begin{cases} 1 & \text{if } i \text{ not perfect square} \quad [\,\Delta\Phi > 0 \text{ \& constant} \\ \sqrt{i} & \text{otherwise} \end{cases}$$



$c_i =$ 1 2 3 4 5

Given i,

Let k be largest such that $k^2 \leq i$

$\boxed{\Phi(i) := i - k^2 \geq 0}$ $\Phi(i) = \boxed{i - \underbrace{\lfloor\sqrt{i}\rfloor}_{k}{}^2}$

~~$i \neq k$~~ $i > k^2 \Rightarrow c_i = 1$

$$\hat{c}_i = c_i + \Delta\Phi = 1 + (i - k^2) - ((i-1) - k^2)$$
$$= 1 + 1 = 2$$

$i = k^2$:

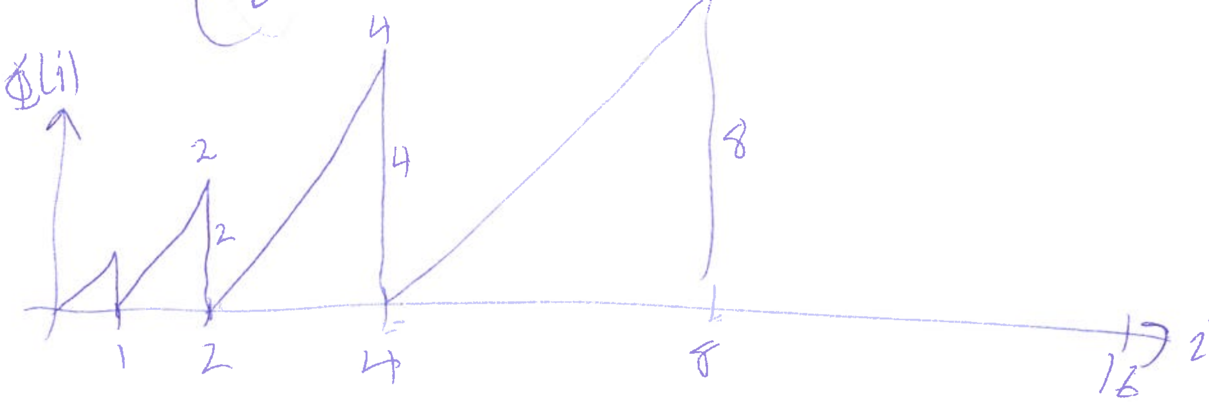
$$\hat{c}_i = c_i + \Delta\Phi = k + \overbrace{(i - k^2)}^{0} - ((i-1) - (k-1)^2)$$
$$= k - i + 1 + k^2 - 2k + 1$$

$= k - \cancel{k^2} + 1 + \cancel{k^2} - 2k + 1$

$= 2 - k \le 1$

$\therefore \hat{c}_i \le 2 = O(1)$

$$c_i = \begin{cases} 1 & \text{if } i \text{ not a power of } 2 \\ i & \text{if } i \text{ is a power of } 2 \end{cases}$$



A2) Let k be biggest such that $2^k \le i$.

$[k = \lfloor \lg i \rfloor]$

$\Phi(i) = 2(i - 2^k)$

$i > 2^k$: $c_i = 1$ and $i - 1 \ge 2^k$

$\hat{c}_i = c_i + \Delta\Phi = c_i + \Phi(i) - \Phi(i-1)$

$= 1 + 2(i - 2^k) - 2(i - 1 - 2^k)$

$= 1 + \cancel{2i} - 2^{k+1} - \cancel{2i} + 2 + 2^{k+1}$

$= 3$

$i = 2^k$: $c_i = i = 2^k$, $\Phi(i) = 0$, $\Phi(i-1) = 2(i - 1 - 2^{k-1})$

$$\hat{c}_i = 2^k + 0 - 2(i-1) - 2^{k-1})$$

$$= 2^k - 2i + 2 + 2^k$$

$$= 2^k - 2^{k+1} + 2 + 2^k$$

$$= 2$$

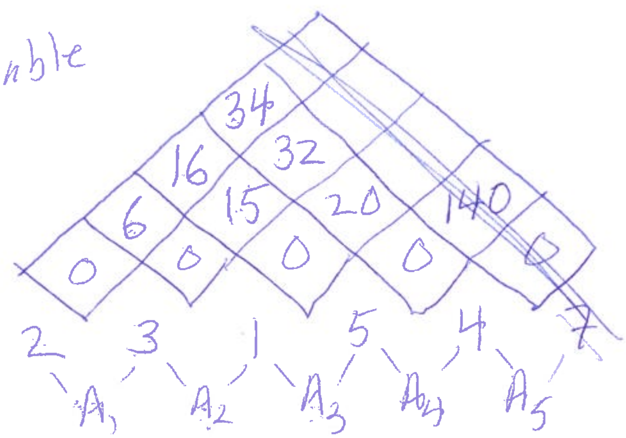
$$2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$$

FIFO Queue w/ 2 stacks (deferred)

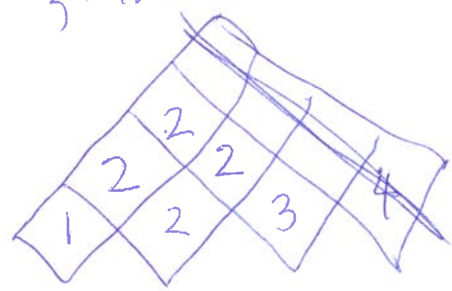
Matrix chain order

$\langle 2, 3, 1, 5, 4, \cancel{7} \rangle$

m-table



s-table



$6 + 2 \cdot 1 \cdot 5 = 16$

$15 + 30 = 45$

$15 + 3 \cdot 5 \cdot 4 = 75$

$20 + 3 \cdot 1 \cdot 4 = 32$

$16 + 2 \cdot 5 \cdot 4 = 56$

$\rightarrow 6 + 20 + 2 \cdot 1 \cdot 4 = 34$

$32 + 2 \cdot 3 \cdot 4 =$

Answer = 34

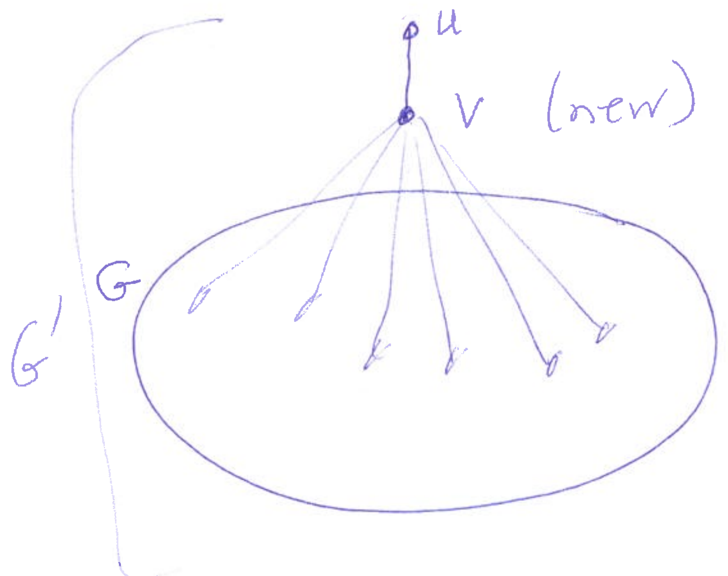
optimal split

$A_{1 \cdots 2} \, A_{3 \cdots 4}$

$= (A_1 A_2)(A_3 A_4)$

$VC \leq_p CON\text{-}VC$

Given $\langle G, k \rangle$, produce in p time $\langle G', k' \rangle$ such that G has a v.c. of size k iff G' is connected and has a v.c. of size k'.



G':

- Start with G
- Add new vx v,
- add edge connecting v with every vertex in G.
- Add vertex u and edge between u and v.

$k' := k+1$

Argue: this clearly p-time

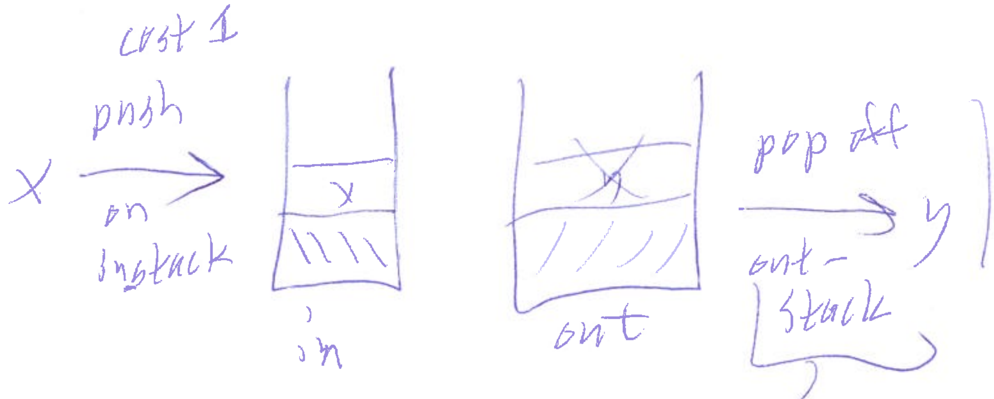
(G' always connected)

G has v.c of size $k \Rightarrow G'$ connected with v.c. of size $k+1$ (just add v to the cover)

G' has v.c. of size $k+1$ & is connected $\Rightarrow G$ has a v.c. of size k

[slight technicality when $k = |G.V|+1$]

Queue with 2 stacks:



cost 1 if out-stack not empty, or |in-stack|+1 otherwise

if out-stack empty,

pop each item of in-stack

& push it onto the out-stack.

pop off out-stack

$\Phi = 2 \cdot |\text{in-stack}|$

const amortized time per op.