Pseudopolynomial

Thursday, April 2, 2020 2:34 PM

COUNT-DIVISORS(13)

1101

4 bits

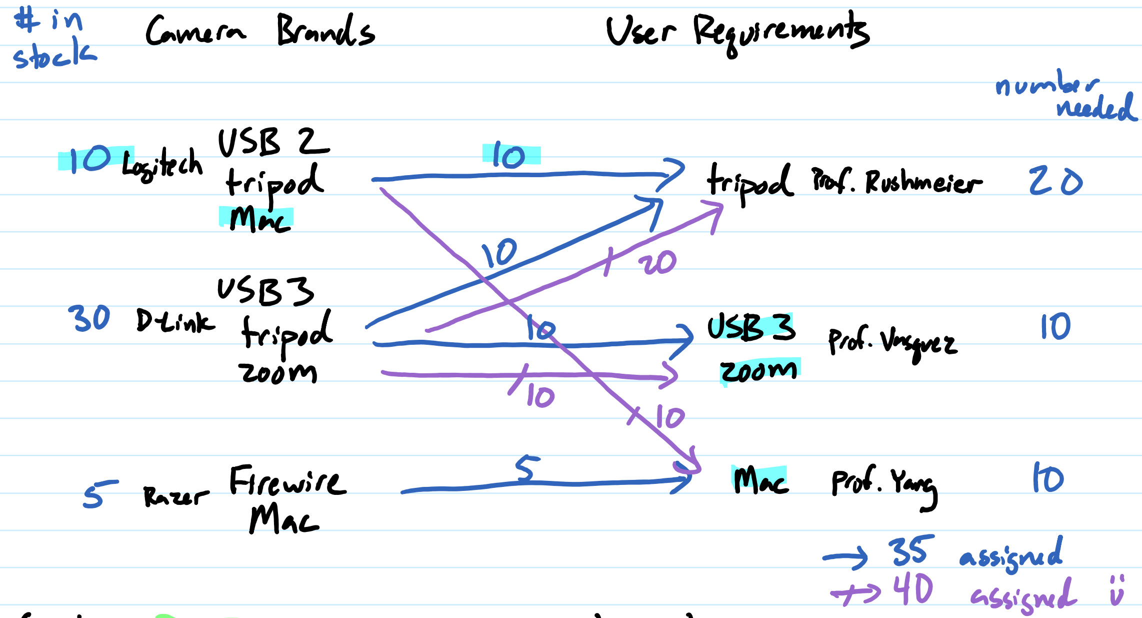
COUNT-DIVISORS(n)

```
count ← 0
for i = 1 to n          ← O(n) iterations
    if n % i == 0
        count ← count + 1    ] O(1)
return count
```

O(n) total

↪ $O(2^{\#\text{ bits input}})$ exponential

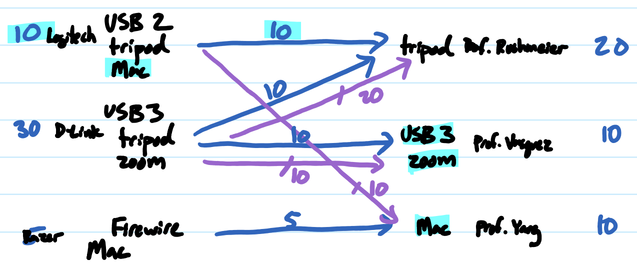
pseudopolynomial!

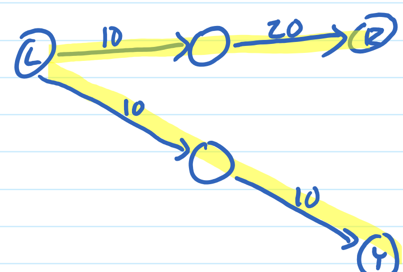


Goal: maximize # cameras sent out

subject to constraints

1) can't assign more of one brand than in stock
2) can't assign more to user than requested
3) can't assign unless have required capabilities





construct G with

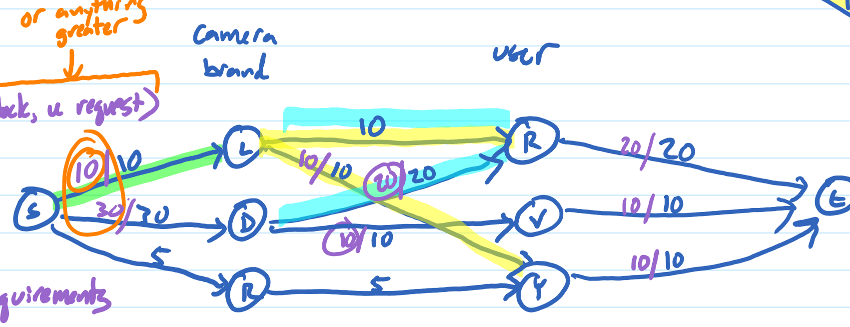
or anything greater

$c(b,u) = \min(b \text{ in stock}, u \text{ request})$

$c(s,b) = b$ in stock

$c(u,t) = u$'s request

only have an edge if brand b satisfies u's requirements



→ $f^{out}(L) = f^{in}(L) = f(s,L) \leq c(s,L) = 10$

in general, for any flow, $f^{out}(b) \leq$ # b available

→ $f^{in}(V) = f^{out}(V) = f(V,t) \leq c(V,t) = 10$

$f^{in}(u) \leq$ # u requested

cameras distribute $= \sum_b$ # brand b distributed

$= \sum_b \sum_u$ # brand b distributed to user u

cameras distributed

$=$

$$= \sum_b \sum_u \text{\# brand } b \text{ distributed to user } u$$

$$= \sum_b \underline{\sum_u f(b,u)}$$

$$= \sum_b f^{out}(b)$$

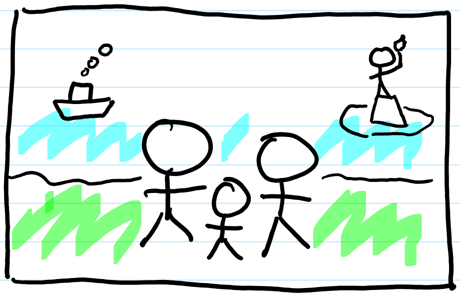
$$= \sum_b f^{in}(b) = \sum_b f(s,b) = \underline{v(f)}$$

#cameras distributed
=
value of flow

so max flow gives
max distribution

(solve with Scaling
Ford-Fulkerson)

Background Segmentation



Determine what pixels are background, which are foreground.

For each, give $a_i = \underline{P(\text{pixel } i \text{ in fore})}$ $\quad b_i = P(\text{pixel } i \text{ in back})$ $\quad a_i + b_i = 1$

consult image processing experts for how to get these – measure blur?

For neighboring pairs, give p_{ij} = penalty for separating pixel i from j (one in fore, one in back)

Find partition $\overset{\text{fore}}{A}, \overset{\text{back}}{B}$ to maximize $\displaystyle\sum_{i\in A} a_i + \sum_{i \in B} b_i - \sum_{\substack{i,j \text{ adjacent} \\ i\in A \text{ xor } j \in A}} p_{ij}$

$$= \sum_{i\in A}(1-b_i) + \sum_{i\in B}(1-a_i) - \sum_{\substack{i,j \text{ adjacent} \\ i\in A \text{ xor } j \in A}} p_{ij}$$

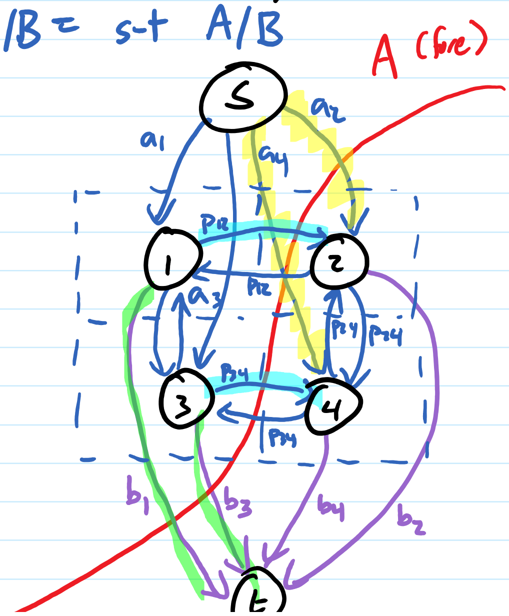
$$= |A| - \sum_{i\in A} b_i + |B| - \sum_{i\in B} a_i - \sum_{\substack{i,j \text{ adjacent} \\ i\in A \text{ xor } j \in A}} p_{ij}$$

$$= \underbrace{Q}_{\substack{\text{total} \\ \#\text{pixels}}} - \sum_{i\in A} b_i - \sum_{i\in B} a_i - \sum_{\substack{i,j \text{ adjacent} \\ i\in A \text{ xor } j \in A}} p_{ij}$$

minimize $\displaystyle\sum_{i\in A} b_i + \sum_{i\in B} a_i + \sum_{\substack{i,j \text{ adjacent} \\ i\in A \text{ xor } j \in A}} p_{ij}$

$= c(A,B)$

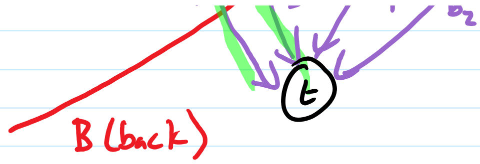
partition A/B = s-t A/B



want to find A, B to minimize capacity of s-t cut (A, B)

$\Downarrow$

Ford-Fulkerson (max-flow $\Leftrightarrow$ min cut)


t
B (back)