COL 351: Analysis and Design of Algorithms

Lecture 38

Quiz 3 (Duration: ~~20~~ 25 min)

Ques 1: Let A and B be two sets, each having n integers in the range $[1,10n]$. The Cartesian sum of A and B is $C = \{x + y \mid x \in A,\ y \in B\}$.

We want to find the elements of C and the number of times each element of C is realized as a sum of elements in A and B. Design an $O(n \log n)$ time algorithm to achieve this objective [**5 marks**].

Ques 2: Let G be a directed graph with unit edge capacities, s be a source, t be a destination. Present a linear time algorithm to verify if (s, t)-max flow in G is bounded by nine [**2.5 marks**].

OR

Let G be a directed graph with unit edge capacities, s be a source, t be a destination. Design an $O(mn)$ time algorithm to verify if G has a unique (s, t)-min-cut [**5 marks**].

n = number of vertices

m = number of edges

Ques 1: Let A and B be two sets, each having n integers in the range $[1,10n]$. The Cartesian sum of A and B is $C = \{x + y \mid x \in A,\ y \in B\}$.

We want to find the elements of C and the number of times each element of C is realized as a sum of elements in A and B. Design an $O(n \log n)$ time algorithm to achieve this objective [**5 marks**].

Solution: Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be input sets.
Define

$A(x) = \sum_{i=1}^{n} x^{a_i}$ and $B(x) = \sum_{j=1}^{n} x^{b_j}$. We can fine $C(x) = A(x) \cdot B(x)$ in $O(n \log n)$ time, because $\deg(A(x)), \deg(B(x)) \leq 10n$

Now suppose $i_1\ i_2 \ldots i_k$ satisfy $a_{i_r} + b_{i_r} = z$, for $1 \leq r \leq k$.

Then, x^z will have co-efficient as k, because it will be obtained by multiplying k terms in $A(x)$ with appropriate terms in $B(x)$

Ques 2(a): Let G be a directed graph with unit edge capacities, s be a source, t be a destination. Present a linear time algorithm to verify if (s, t)-max flow in G is bounded by nine [**2.5 marks**].

$f = 0$

For ($i = 1$ to 10):

If $\exists$ an (s,t) path (say P) in G_f:

→ If $i = 10$: Return "Not bounded by 9"

→ Update f by increasing max possible flow along path P.

Else

Return "Bounded by 9".

$\text{Time} = O(10(m+n)) = O(m+n)$

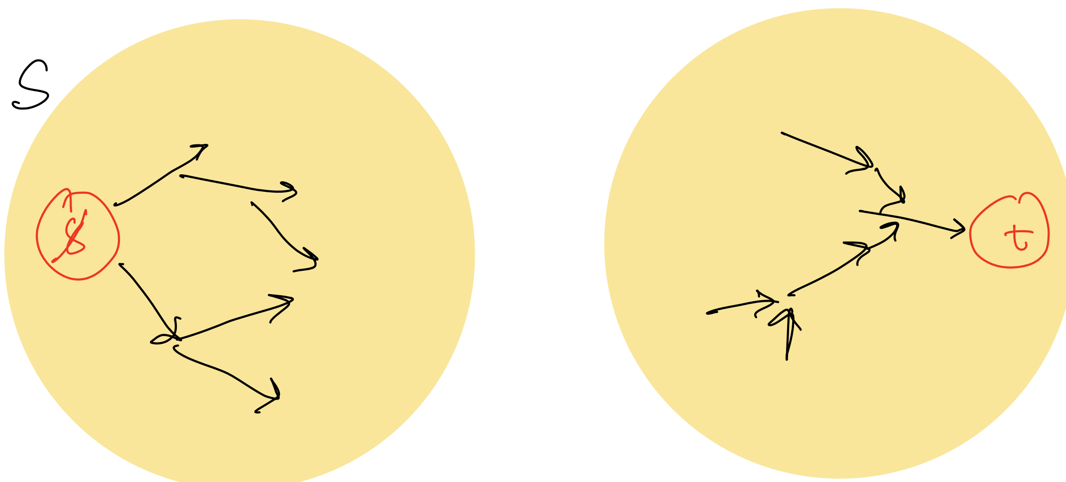
Ques 2(b): Let G be a directed graph with unit edge capacities, s be a source, t be a destination. Design an $O(mn)$ time algorithm to verify if G has a unique (s,t)-min-cut [**5 marks**].

Let $f \equiv$ any (s,t)-max-flow

$S =$ vertices reachable from s in G_f

$T =$ vertices reachable to t in G_f

Time $= O(m \cdot n)$



Part 1: If $S \cup T \subsetneq V(G) \implies \exists$ two (or more) min-cuts

Proof: (S, S^c), (T^c, T) — These are 2 min-cuts.

Part 2: If $S \cup T = V(G) \implies \exists$ a unique min-cut

> **CLAIM:** For any (s,t)-min-cut (X, Y) we have $S \subseteq X$, $T \subseteq Y$
>
> **Proof:** If $\exists$ a vertex $w \in S \cap Y$, then it will imply there is an edge $(x,y) \in E \cap (X \times Y)$ in residual graph G_f, corresponding to (s,t)-max-flow f. This is not possible as (s,t)-max-flow value $\equiv$ (s,t)-min-cut-value.

Now if $S \cup T = V(G)$, then the only possiblity for any min-cut (X, Y) is $S = X$ and $T = Y$, thus implying uniqueness.