COL 702

Homework IV

Due on Nov. 14, 2021

Justify your answers with proper reasonings/proofs.

1. Suppose you are given an array $M[1...n, 1...n]$ of numbers, which may be positive, negative, or zero, and which are not necessarily integers. Describe an algorithm to find the largest sum of elements in any rectangular subarray of the form $M[i...i', j...j']$ Your algorithm should run in $O(n^3)$ time.

   Solution: For each entrie $M[i,j]$ and index l, we will compute the sum of entries $M[i,j] + M[i,j-1] + ... + M[i,j-l+1]$, i.e., we consider row i, and compute the sum of the l entries before $M[i,j]$ in this row. We store this value in a table $T[i,j,l]$. Note that all these entries can be computed in $O(n^3)$ time :

   $$T(i,j,l) = M[i,j] + T(i,j-1,l-1).$$

   Now, for each matrix entrie $M[i,j]$ and paramter l, let $A(i,j,l)$ denote the maximum sum array whose width is l and the bottom right corner is $M[i,j]$. For computing $A(i,j,l)$, there are two choices – either this subarray contains only one row (row i) or it contains row $(i-1)$ as well. So

   $$A(i,j,l) = \max(T(i,j,l), T(i,j,l) + A(i-1,j,l)).$$

2. For a sequence of n days, you are given subsets $S_1, S_2, \ldots, S_n$ of $\{1, 2, \ldots, k\}$. Think of S_i as the subset of people who are available for work on day i. You need to pick exactly one person from S_i for each of the days $i = 1, \ldots, n$. For a person j, let Δ_j denote the $\sum_{i:j \in S_i} \frac{1}{|S_i|}$. This is the expected number of times j would be picked if we pick a random person from S_i on each of days $i = 1, \ldots, n$. A selection of persons, one from each set S_i, is said to be good if each person j is picked at most $\lceil \Delta_j \rceil$ times. Show that such a selection is always possible, and give an efficient algorithm to find such a selection.

   Solution: We construct a directed graph as in the case of bipartite matching. First construct a bipartite graph H as follows: on left side L, we have one vertex for each of the days $1, ..., n$, call these $v_1, \ldots, v_n$. On the right side R, we have one vertex w_j for each person j. For every $v_i \in L$ and person $w_j \in R$, add an edge between v_i and w_j iff $j \in S_i$. Now direct all these edges from L to R, and assign infinite capacities to them. Finally, add a new vertex s to and a new vertex t. We have an edge from s to each vertex in L with capacity 1, and an edge from each vertex w_j in R to t with capacity $\lceil \Delta_j \rceil$. Now the required solution exists if and only if there is a flow of value n in this graph, i.e., every cut has capacity at least n.

   So now let X be an s-t cut. Assume there is no infinite capacity edge leaving X, otherwise it has infinite capacity. Let L_1, R_1 be the vertices in L and R which are present in X respectively. Note that every edge leaving a vertex in L_1 lies in R_1 (otherwise we have an infinite capacity edge leaving X). Now, the capacity of the cut X is

$$\begin{aligned}\sum_{v_i \notin L_1} 1 + \sum_{w_j \in R_1} \lceil \Delta_j \rceil &= n - |L_1| + \sum_{w_j \in R_1} \lceil \Delta_j \rceil \geq n - |L_1| + \sum_{w_j \in R_1} \sum_{i : j \in S_i} \frac{1}{|S_i|} \\ &= n - |L_1| + \sum_i \sum_{j : w_j \in R_1, j \in S_i} \frac{1}{|S_i|} \geq n - |L_1| + \sum_{i \in L_1} \frac{|\{j : w_j \in R_1, j \in S_i\}|}{|S_i|}\end{aligned}$$

But for each $i \in L_1$, all its neighbours in R are present in R_1. Therefore, the set $\{j : w_j \in R_1, j \in S_i\}$ is same as S_i. In other words, the RHS above is at least $n - |L_1| + |L_1| = n$, and so the min-cut is at least n.

3. Let (u, v) be a directed edge in arbitrary flow network G. Prove that if there is a minimum (s, t)-cut (S, T) such that $u \in S$ and $v \in T$, then there is no minimum cut (S', T') such that $u \in T', v \in S'$). Note that by definition of cut, $s \in S, t \in T$, and similarly $s \in S', t \in T'$.

   Solution: Let f be a max-flow. Note that a cut (S, T) is a min-cut if and only if for all edges $e \in \delta^+(S)$, $f_e = u_e$, and for all edges $e \in \delta^-(S)$, $f_e = 0$. This is because for any cut (S, T), the value of the flow is equal to the total flow leaving S minus the total flow entering S. So if e denotes (u, v), then $f_e = u_e$. But then if $e \in \delta^-(S')$ for a cut (S', T'), then the total flow entering S' is positive, and so this cannot be a min-cut.

4. Let G be an undirected graph and s and t be two special vertices in it. Give an efficient algorithms to find the maximum number of node disjoint paths from s to t (a set of paths from s to t are said to be node-disjoint if no two of them share a vertex other than s or t).

   Solution: We construct a new directed graph H from G as follows. For every vertex v in G, there are two vertices v' and v'' in H and we add an edge from v' to v'' in H. If (u, v) is an edge in G, then we add directed edges (u'', v') and (v'', u') to H.

   Let $s = v_0, v_1, ..., v_k = t$ be an s-t path in G. Then we get a path $v_0', v_0'', v_1', v_1'', v_2', v_2'', ..., v_k', v_k''$ in H. Now notice that if there are two paths in G from s to t which are vertex disjoint, then the corresponding paths in H will be edge disjoint, and conversely. So we need to find the maximum number of edge disjoint paths in H from v_0' to v_k''. But we know how to solve this problem (using max-flow, done in class).

5. Let G be an undirected graph. For a subset S of vertices, let $e(S)$ denote the number of edges which have both the end-points in S. Given a rational number α, we would like to find out if there us a subset S of vertices such that $\frac{e(S)}{|S|} \geq \alpha$. Give an efficient algorithm to solve this problem.

   Solution: We need to find a set of vertices S such that $e(S) - \alpha|S| \geq 0$. Define a new graph H as follows. For every vertex v in G there is a vertex in G – call it x_v. For every edge $e \in G$, there is also a vertex x_e in H. Also there are two more vertices s and t in H. Add an edge from s to x_e of capacity 1 for every edge e of G, and an edge from x_v to t of capacity α for every vertex v in G. Further if e is an edge incident with v in G, then add an edge from x_e to x_v with infinite capacity. Now we claim that there is a subset S of vertices in G with $e(S) - \alpha|S| \geq 0$ if and only if the minimum s-t cut in H is at most m, where m is the number of edges in G. To see this, let T be an s-t cut in H of finite capacity. Let T_V denote the set of vertices in T which are of the form x_v and T_E be the vertices of the form x_e. Notice that if $x_e \in T_E$, then $x_v \in T_V$, where v is end-point of e. Therefore, let S denote the set of vertices which happen to be one of the end-points of an edge in T_E. Then S is contained in T_V, and so, $|T_E| \leq e(S)$. Now the capacity of this cut is

$$(m - |T_E|) + \alpha|T_V| \geq m - (e(S) - \alpha|S|).$$

So if min-cut is at most m, then there is a set S with $e(S) - \alpha|S| \geq 0$, and this can be found by a min-cut computation on H. Conversely, suppose there is a subset S of vertices in G with $e(S) - \alpha|S| \geq 0$, then consider the following cut T in H: T_V is S and T_E consists of edges with both end-points in S. As above, the capacity of this cut T is at most m.