Algorithms: Assignment sheet 2

Due date: October 31, 2025

---

1. We want to find a global min-cut $(A, V \setminus A)$ in an undirected graph $G = (V, E)$ (with unit edge weights and $|V| = n$) such that $|A|$ is the smallest among all global min-cuts in G. Show a randomized polynomial time algorithm with running time $O(n^2 \cdot \text{poly} \log n)$ that finds such a global min-cut in G with probability at least 3/4.

2. Let $G = (V, E)$ be an undirected graph. For any $X \subseteq V$, let $\delta(X)$ denote the size of the cut $(X, V \setminus X)$, i.e., the number of edges with one endpoint in X and another endpoint in $V \setminus X$. If A and B are any two subsets of V, show that $\delta(A) + \delta(B) \geq \delta(A \cap B) + \delta(A \cup B)$.

3. Prove that following statement on min cuts in G: if $(S, V \setminus S)$ is a minimum s-t cut in G and $u, v \in S$, then there exists a minimum u-v cut $(U, V \setminus U)$ such that $U \subset S$ or $V \setminus U \subset S$.

4. Consider the following generalization of a matching called a "fractional matching": h is a fractional matching in a bipartite graph $G = (A \cup B, E)$ if h is a function from the edge set E to $[0, 1]$ such that for every vertex $u \in A \cup B$, we have $\sum_e h(e) \leq 1$, where the sum is over all edges e incident on u. A fractional matching h is A-perfect if the above sum exactly equals 1 for each $u \in A$. What is the analogue of Hall's theorem for a bipartite graph G to admit an A-perfect fractional matching?

5. Show that the running time of the algorithm to compute a maximum cardinality matching in bipartite graphs can be improved to $O(m\sqrt{n})$ by computing max flow using Dinic's algorithm. (m is the number of edges and n is the number of vertices)

6. Let $G = (V, E)$ be a directed graph with edge weight function $w : E \to R$. Thus negative edge weights are allowed here, however let us assume there are no negative-weight cycles. Let $|V| = n$ and let $s \in V$ be the source vertex.

   Show that the following algorithm correctly computes shortest paths and distances from s. Start with $d[s] = 0$ and $d[u] = \infty$ for all $u \in V - \{s\}$ and $\pi(v) = \text{nil}$ for all $v \in V$.

   – for $i = 1, \ldots, n - 1$ do
      – for each edge $(u, v) \in E$ do: *relax*(u, v).

   – Return the arrays π and d.

   [The subroutine *relax*(u, v) is the same as the one we saw in Dijkstra's algorithm.]

7. Suppose the above directed graph $G = (V, E)$ has a negative-weight cycle that is reachable from the source s. Give an efficient algorithm to list the vertices of such a cycle.

8. The following are Fibonacci-heap operations: *extract-min*$(\cdot)$, *decrease-key*$(\cdot, \cdot)$, and also *create-node*(x, k) which creates a node x in the root list with key value k. Show a sequence of these operations that results in a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.

9. Let T be a minimum spanning tree of a graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree T' of G, the list L is also the sorted list of edge weights of T'.

10. *Boruvka's algorithm.* Assume all edge weights are distinct in an undirected connected graph $G = (V, E)$ on m edges and n vertices. Consider the following step: we simultaneously contract the minimum weight edge incident on every vertex.

    (a) Show that this step can be performed in $O(m)$ time.

    (b) Show that the resulting graph has at most $n/2$ vertices.

    (c) Show that the MST of G is the union of the edges marked for contraction and the edges of the MST of the resulting graph.