Algorithms: Assignment sheet 2

Due date: November 20, 2020

---

1. 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 $\delta(A) + \delta(B) \geq \delta(A \cap B) + \delta(A \cup B)$.

2. 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 $V \setminus U \subset S$.

3. 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?

4. Given an undirected graph $G = (V, E)$, the edge colouring problem is an assignment of the smallest number of colours to the edges so that no two edges incident on the same vertex have the same colour. Show that there is an edge colouring of any d-regular bipartite graph G (every vertex in G has degree d) by d colours.

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. Suppose that we are given a weighted directed graph $G = (V, E)$ in which edges that leave the source vertex s may have negative weights, all other edge weights are non-negative, and there are no negative-weight cycles. Show that Dijkstra's algorithm correctly finds shortest paths from s in this graph.

7. 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.]

8. 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.

9. Let us modify the "cut rule" (in the implementation of decrease-key operation for a Fibonacci heap) to cut a node x from its parent as soon as it loses its 3rd child. Recall that the rule that we studied in class was when a node loses its 2nd child. Can we still upper bound the maximum degree of a node of an n-node Fibonacci heap with $O(\log n)$?

10. 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.