Home Exam (example)

Remarks: All the graphs here are without self loops and parallel or anti-parallel edges. In all the algorithms, always explain their correctness and analyze their complexity. The complexity should be as small as possible. A correct algorithm with large complexity, may not get full credit. The number of vertices is denoted by n, and the number of edges by m.

Choose 5 out of the next 6 questions.

Question 1: Give an algorithm that gets as input a *sorted* array A of positive integers. Assume that the integers are pairwise distinct (no value appears more than once). Give an algorithm that checks if there is an index i so that $A[i] = i$. You may assume for simplicity that n is a power of 2.

Answer: Say that $A[n/2] > n/2$. In this case, as the elements are distinct and the array is sorted, $A[n/2+1] > A[n/2] \geq n/2+1$. Namely, $A[n/2+1] > n/2+1$. In the same way, you now show that $A[n/2+2] > n/2+2$ and so on.

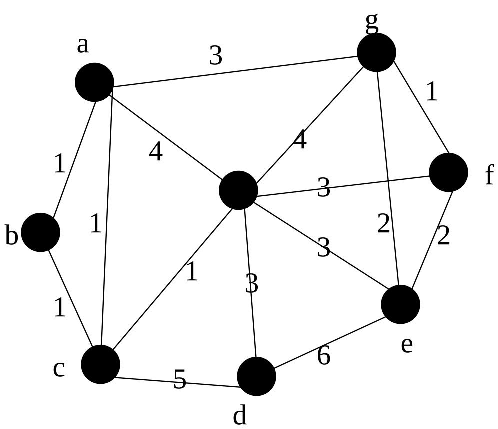
Thus, if $A[n/2] > n/2$, we can look for an i such that $A[i] = i$, in indices $n/2-1$ or lower. In the same way, if $A[n/2] < n/2$, we can look for the required i in $n/2+1$ and higher. And if $A[n/2] = n/2$, we found the required $i = n/2$.

Thus, we can perform a procedure like binary search to find the i. The running time is $O(\log n)$.

Question 2: Give an algorithm that gets as input a directed graph. The algorithm should output the number of pairs (u, v) in the graph, so that there is a path from u to v. Note that (u, v) is not the same as (v, u).

Answer: Use n times the BFS procedure. When doing BFS from v, let P_v be the number of vertices that are reachable from v (there is a path from v to them). This is exactly the number of vertices that are given a label in the BFS run (so their label is not infinite). This gives all the pairs of the form $(v, \ )$. The required answer is $\sum_v P_v$. The time complexity is $O(mn)$.

Question 3: Run the Kruskal algorithm on the following graph:



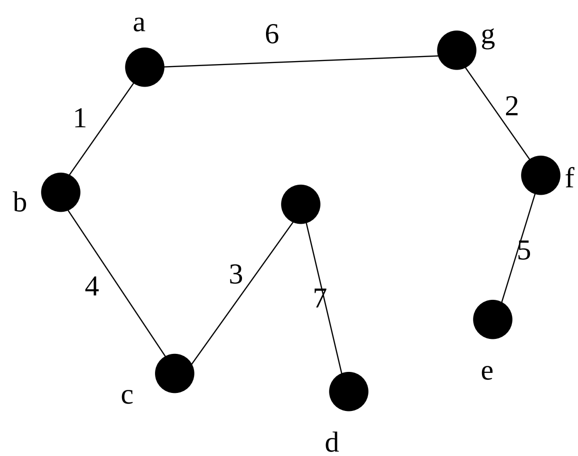


Figure 1: Te run of the Kruskal algorithm

All you need to do is copy the vertices and the tree edges only. On the edges you write a number between 1 and 7, representing the order by which the edge is added into the solution.

Question 4: A tournament graph is a *directed* graph with all the $(\binom{n}{2})$ edges present, each with a single direction (out of the two possible ones). An Hamiltonian path is a simple path of length $n-1$, that contains all the vertices.

1. Prove that every tournament graph has an Hamiltonian path.
   Answer: We prove by induction with the basis of induction $n = 2$ clear. Say this is true for n and consider a graph with $n+1$ vertices. Remove a vertex x from the tournament. We are left with a tournament G_n of n vertices. By the induction hypothesis, there is an Hamiltonian path $v_1 \to v_2 \to, \ldots, v_n$ in G_n. Let v_i be the rightmost vertex in the Hamilton path such that the edge is $v_i \to x$ and not $x \to v_i$. So, for v_{i+1}, the edge must be $x \to v_{i+1}$. So we can insert x between v_i and v_{i+1} and get an Hamiltonian path in G.
2. Give an algorithm that finds an Hamiltonian path in a tournament.
   Answer: Choose an arbitrary x and remove it. Recursively find an Hamiltonian path in $G \setminus \{x\}$. Find v_i as explained above and insert x after v_i. The complexity: per each vertex we have $O(V)$ search, so $O(V^2)$.

Question 5: Give an algorithm that checks if there is a, (not necessarily simple) paths from v to u of length exactly k.
Answer: If there is a path between v and u of length k, then u is a neighbor of a vertex w for whom there is a path from v of length $k-1$. So the algorithm is as follows:

1. $S_0 = \{v\}$
2. For i=1 to k do

   Let S_i be the set of vertices w with a vertex $z \in S_{i-1}$, and $z \to w \in E$.
3. Check if $u \in S_k$

This takes $O(E)$ for every i (go over all the edges maybe) so $O(E \cdot k)$.

Question 6: A DFS procedure was runed on a graph G and we got that $\sum_{v \in V} Low(v) = |V|$. We also know that the number of leaves in the DFS tree is l.

1. Prove that there are no more than l bio-connected components in G.
2. Show an example for the above, with G having exactly l bio-connected components.

Answer: Will be given in class.