1. **Consider a connected undirected graph with distinct edge costs. Which of the following are true? [Check all that apply.]**
2. **You are given a connected undirected graph G with distinct edge costs, in adjacency list representation. You are also given the edges of a minimum spanning tree T of G. This question asks how quickly you can recompute the MST if we change the cost of a single edge. Which of the following are true? [RECALL: It is not known how to deterministically compute an MST from scratch in O(m) time, where m is the number of edges of G.] [Check all that apply.]**
3. **Which of the following graph algorithms can be sped up using the heap data structure?**
4. **Which of the following problems reduce, in a straightforward way, to the minimum spanning tree problem? [Check all that apply.]**
5. **Recall the greedy clustering algorithm from lecture and the max-spacing objective function. Which of the following are true? [Check all that apply.]**
6. **We are given as input a set of n jobs, where job j has a processing time pj and a deadline dj. Recall the definition of completion times Cj from the video lectures. Given a schedule (i.e., an ordering of the jobs), we define the lateness lj of job j as the amount of time Cj−dj after its deadline that the job completes, or as 0 if Cj≤dj.**

**Our goal is to minimize the total lateness, ∑jlj.**

**Which of the following greedy rules produces an ordering that minimizes the total lateness?**

**You can assume that all processing times and deadlines are distinct.**

1. **Consider an alphabet with five letters, {a,b,c,d,e}, and suppose we know the frequencies fa=0.28, fb=0.27, fc=0.2, fd=0.15, and fe=0.1. What is the expected number of bits used by Huffman’s coding scheme to encode a 1000-letter document?**

Solution : 2250

1. **Which of the following extensions of the Knapsack problem can be solved in time polynomial in n, the number of items, and M, the largest number that appears in the input? [Check all that apply.]**
2. **Consider an instance of the optimal binary search tree problem with 7 keys (say 1,2,3,4,5,6,7 in sorted order) and frequencies w1=.2,w2=.05,w3=.17,w4=.1,w5=.2,w6=.03,w7=.25. What is the minimum-possible average search time of a binary search tree with these keys?**

Solution : 2.23