Purdue University S. Hambrusch - C. Guerra
CS 381 Fall 1999

Assignment 8

Due: Monday, Nov. 22, 1999, in class

1) (10 pts.) Prove or disprove each of the following claims.
(i) Let G be a weighted, directed graph in which edge weights can be negative. Dijkstra's algorithm for solving the Single-Source-Shortest-Paths problem works correctly on G, provided that G contains no negative-weight cycles.

(ii) Let G be a weighted, directed graph in which some edge weights are negative. Let G' be the graph obtained from G by adding to every edge weight the absolute value of the minimum weight among all edges in G. From the Single-Source-Shortest-Paths tree for G' the Single-Source-Shortest-Paths tree for G can easily be generated.

2) (10 pts) Let G be a weighted, undirected graph with positive edge weights.
(i) Give an example of such a graph G and a vertex v such that the minimum-cost spanning tree of G is the same as the shortest path tree rooted at v.
(ii) Give an example of of such a graph G and a vertex v such that the minimum-cost spanning tree of G is very different from the shortest path tree rooted at v. Can the two trees have disjoint edge sets?

3) (15 pts) The Tippe Outpost Company maintains n posts along the Wabash River. At any of these posts you can rent a canoe which can be returned at any other post downstream. (It is next to impossible to paddle against the current.) For each possible departure point i and each possible arrival point j, $c(i,j)$ represents the cost of a rental between i and j. It is possible that the cost of renting from i to j is higher than the total cost of a series of shorter rentals. In this case you can return the canoe at some intermediate post k between i and j and continue the trip in another canoe. There is no extra charge for changing canoes.

(i) Describe an dynamic programming algorithm to determine the minimum cost of a trip by canoe from each possible departure point i to each possible arrival point j. Analyze the running time of your algorithm in terms of n.

(ii) Describe the changes to be made to your algorithm so that not only the minimum costs are generated, but the canoe changing locations can be determined, if needed. Then, describe an $O(n)$ time algorithm to determine, for a given pair i and j, the canoe changing locations of a minimum cost trip.

4) (15 pts)
Let $A_1, A_2, .., A_n$ be matrices where the dimensions of A_i are $d_{i-1} \times d_i$ for $i = 1, \cdots, n$. Here is a proposal for a greedy algorithm to determine the best order in which to perform the matrix multiplications to compute $A_1 \times A_2 \times \cdots \times A_n$.
In the following dim is an array containing the dimensions $d_0, \cdots, d_n$

```
greedyOrder(dim,n)
At each step choose the largest remaining dimension
(among dim[1], ..., dim[n-1])
and multiply two adjacent matrices that share that dimension.
```

Observe that this strategy produces the optimal order of multiplications for the matrices in example 10.4 of the text.
(i) What is the order of the running time of this algorithm?
(ii) Either give a convincing argument that this strategy will always minimize the number of multiplications or give an example where it does not do so.