

# **CSC373: Lecture 10**

Discuss problem set

Reflections on DP for least cost paths  
problem

The all pairs least cost problem and  
alternative DP

# A DP with a somewhat different style

- Lets consider the single source least cost paths problem which is efficiently solved by Dijkstra's greedy algorithm for graphs in which all edge costs are non-negative.
- The least cost paths problem is still well defined as long as there are no negative cycles; that is, the least cost path is a simple path.

# Single source least cost paths for graphs with no negative cycles

- Following the DP paradigm , we consider the nature of an optimal solution and how it is composed of optimal solutions to ``subproblems''.
- Consider an optimal simple path  $P$  from source  $s$  to some node  $v$ . This path could be just an edge but if the path  $P$  has length greater than 1, then there is some node  $u$  which immediately precedes  $v$  in  $P$ . If  $P$  is an optimal path to  $v$ , then the path leading to  $u$  must also be an optimal path.
- We are led to define the following semantic array:  
 $C[i, v]$  = the minimum cost of a simple path with path length at most  $i$  from source  $s$  to  $v$ . (If there is no such path then this cost is infinite.)
- The desired answer is then the single dimensional array derived by setting  $i = n-1$  where  $n = |V|$ .

# Corresponding computational array

- $C'[0,v] = 0$  if  $v = s$  and infinite otherwise.
- $C'[i,v] = \min \{A, B\}$  where  $A = C'[i-1,v]$  and  $B = \min \{C'(i-1, u) + c(u,v) \mid (u,v) \text{ in } E\}$ .
- Note: This presentation is slightly different than in the KT text.
- Why is this a slightly different form than before?  
Showing the equivalence between the semantic and computational arrays is not an induction on the number of input items in the solution but is based on some other parameter (i.e. the path length).
- Complexity:  $n^2$  entries and  $O(n)$  time/entry =  $O(n^3)$

# Can we compute the most cost path using the same DP?

- To define this problem properly we want to say “most cost simple path” since cycles will add to the cost of a path. For least cost we did not have to specify that the path is simple.
- Now suppose we just replace min by max in the least cost DP. Namely,  $M[i,v]$  = max cost simple path from  $s$  to  $v$ .
- The corresponding computational array is  
 $M'[0,v] = 0$  (or  $-\infty$ )

$M'[i,v] = \max \{A, B\}$  where  $A = M'[i-1,v]$  and  
 $B = \max \{M'(i-1, u) + c(u,v) \mid u: (u,v) \text{ in } E\}$ . Correct??

# What goes wrong?

- The problem calls for a maximum *simple* path but the recursion

$$B = \max \{M'(i-1, u) + c(u,v) \mid u: (u,v)\} \text{ in } E$$

*does not guarantee that the path through u will be a simple path as v might occur in the path to u.*

*Why isn't this a problem for least cost paths?*