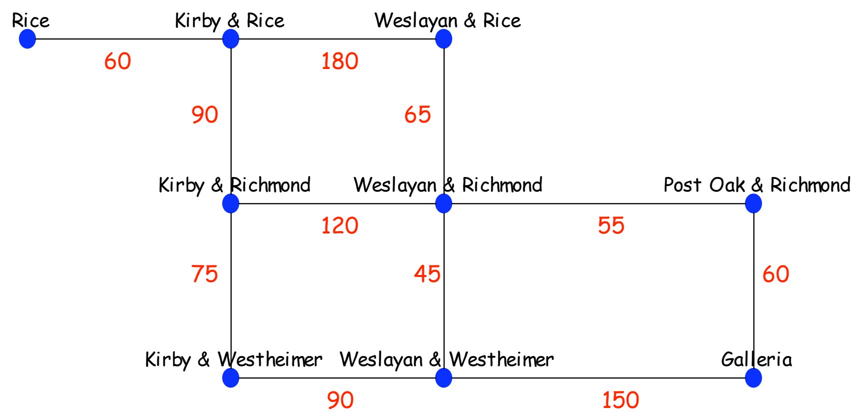
Rice to the Galleria
with dynamic planning

COMP 200, Lecture 19

Rice University

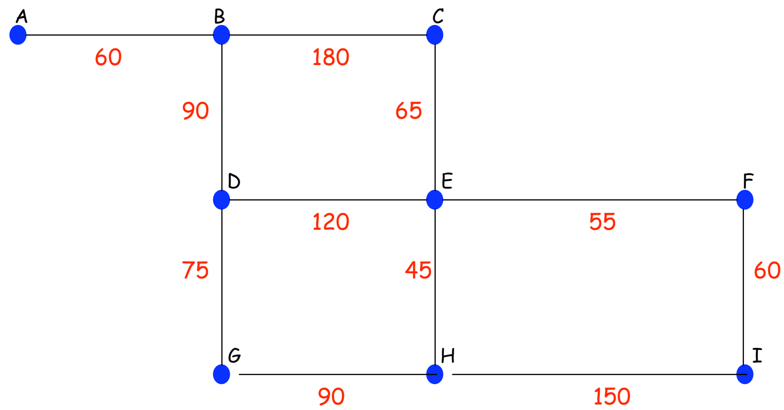
Fall 2004

Rice to Galleria



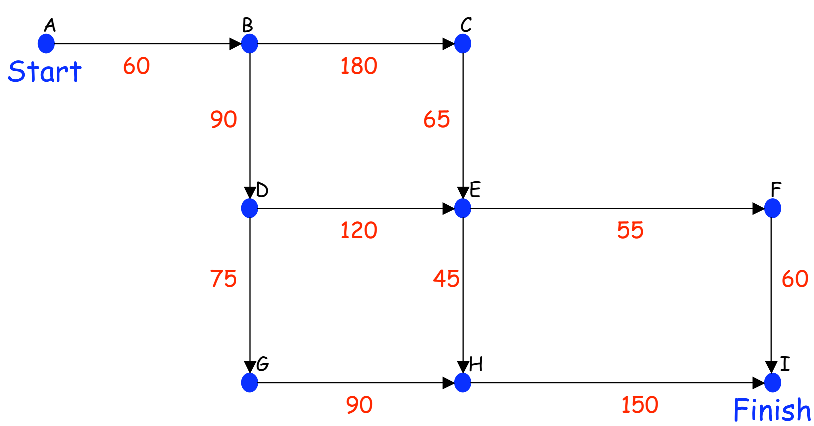
Highly simplified graph of the problem instance

Rice to Galleria



Change the names to protect the innocent

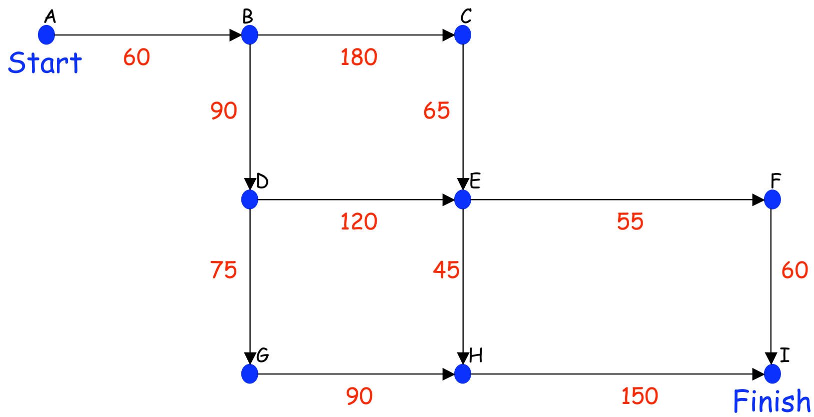
Rice to Galleria



Make all edges directed west or south

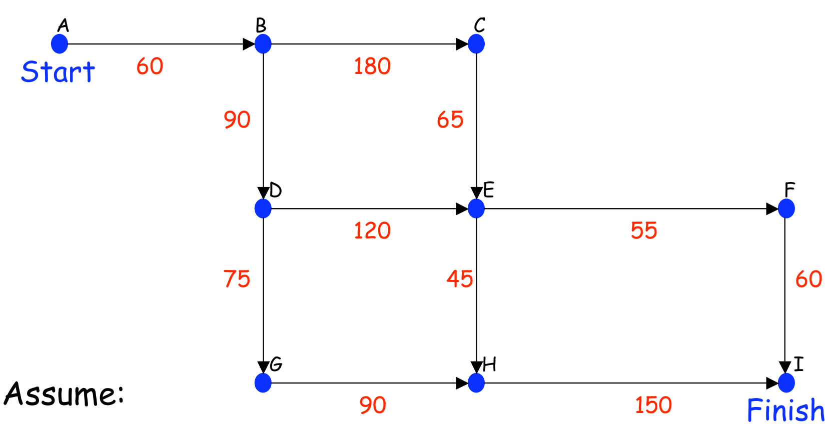
(no cycles)

Rice to Galleria



Find the Optimal Path(s)

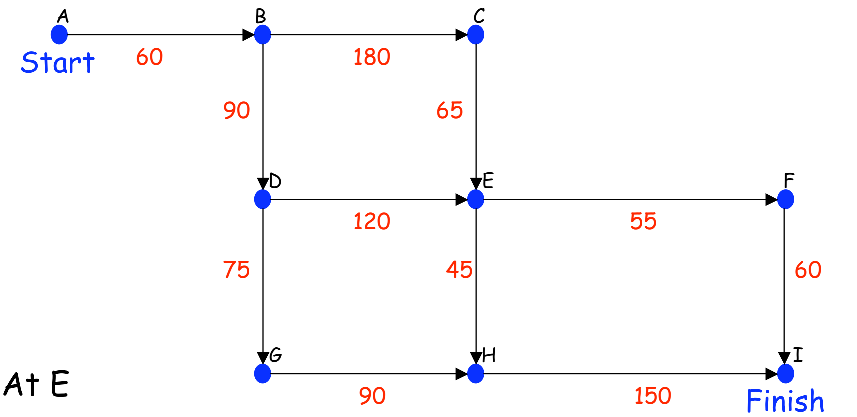
Rice to Galleria



Assume:

- Optimal path can be composed of optimal subpaths
- Compute optimal path from Finish to each node

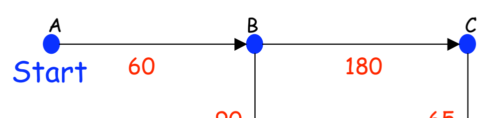
Rice to Galleria

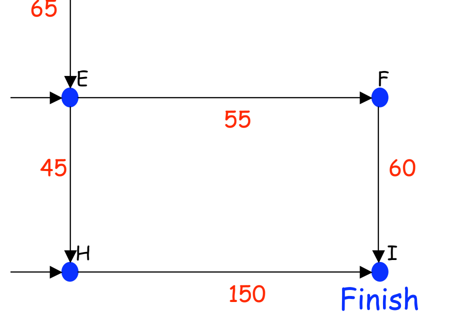


At E

- Optimal path has one of two costs
  - → 45 + cost(H) or 55 + cost(F)
- Given costs for H and F, we can pick optimal path for E

Rice to Galleria





<u>Algorithm is easy</u>

done ← {Finish}

until |done| = nodes in *G*

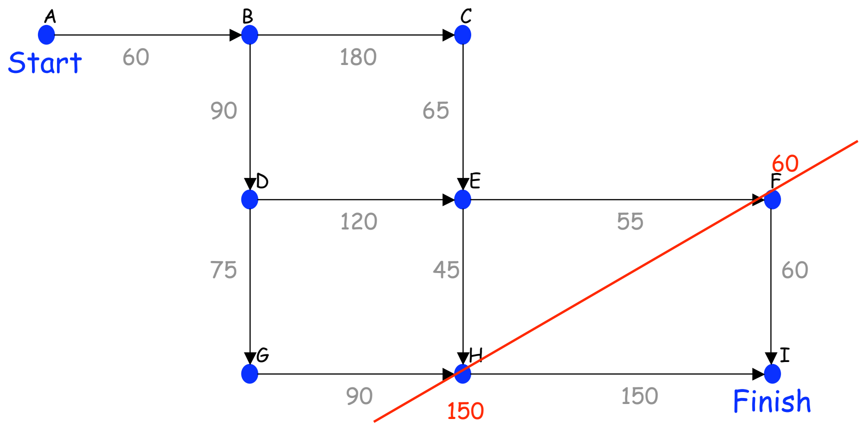
  for each node *n* in *G*

    if every edge leaving n reaches a node in done, compute *n*'s cost and add *n* to done

cost(Start) is the cost of the optimal path

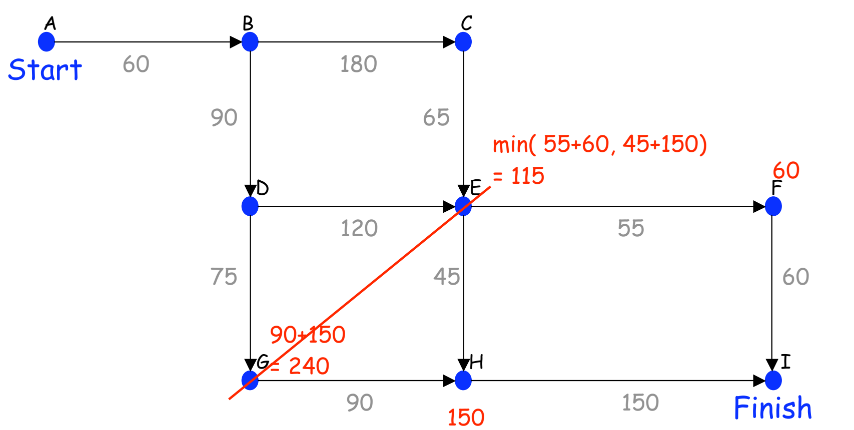
To get the actual path, just remember it as the algorithm builds the costs

Rice to Galleria



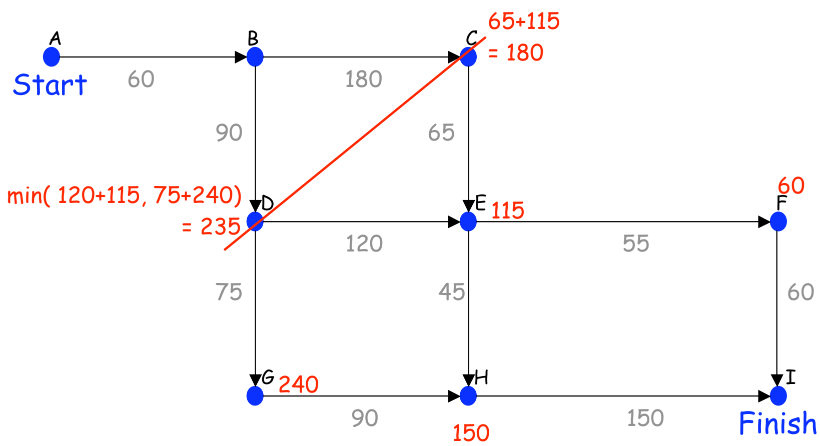
First time through the loop

Rice to Galleria



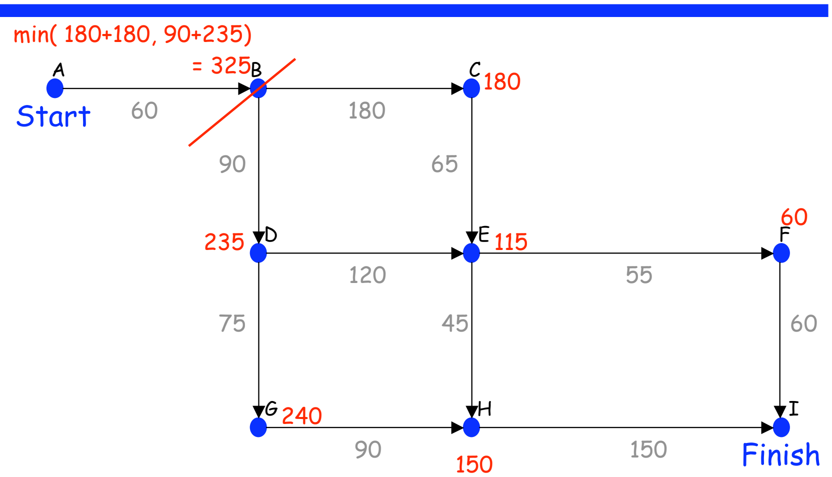
Second iteration

Rice to Galleria



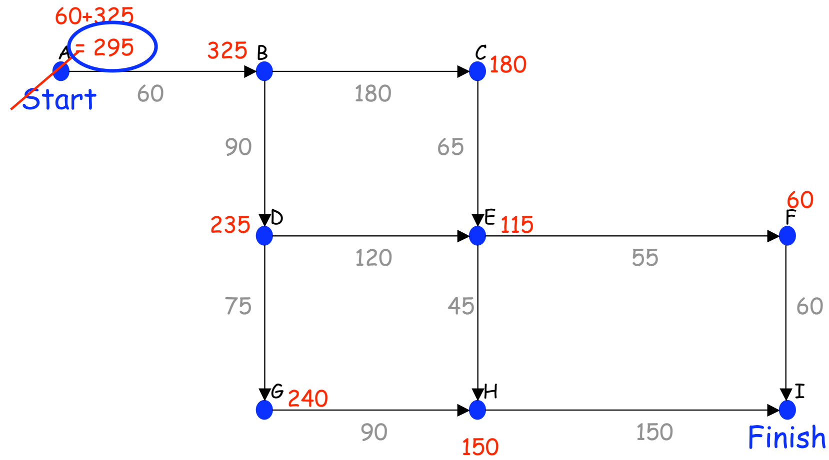
Third iteration

Rice to Galleria



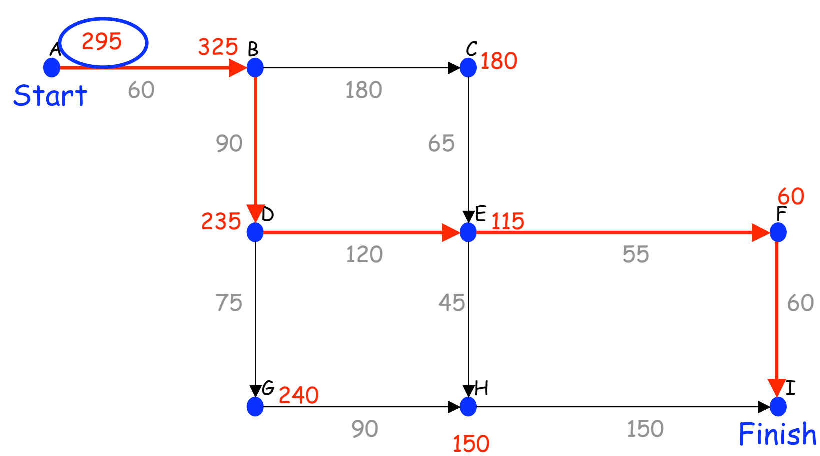
Fourth iteration

Rice to Galleria



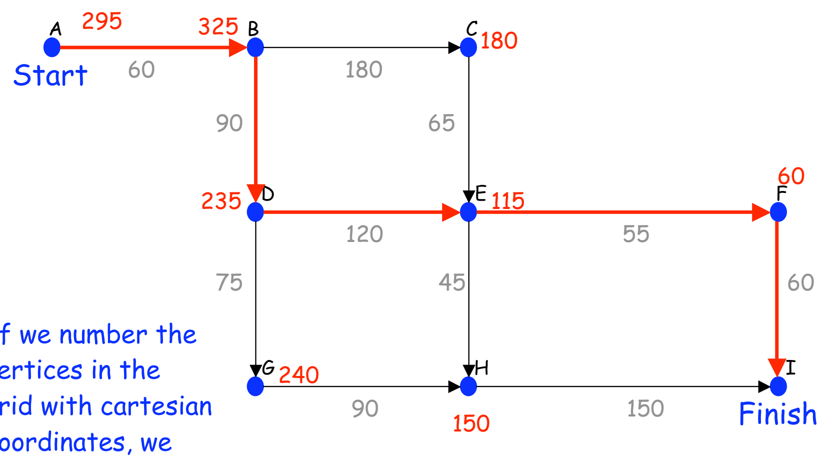
Fifth (& final) iteration

Rice to Galleria



Minimal cost path

Rice to Galleria



Minimal cost path

If we number the vertices in the grid with cartesian coordinates, we can make the tabulation more systematic.

Generalizing From Our Experience

What makes a problem susceptible to dynamic planning?

- Combination of local optima produces global optimum
- Optimality criteria can be formulated as recurrence
  - → c(i,j) = min(cost(<i,j><i,j+1>) + c(i,j+1), cost(<i,j><i+1,j> + c(i+1,j))

The tabulation process, while more expensive than a simple greedy algorithm, is much less expensive than enumerating all the possibilities