Data Structures and Algorithms

UBCO Master of Data Science – DATA 532

Recap

We looked various applications for searching in Graphs

We specifically looked at two alogrithms

- Breadth First Search
- Depth First Search

Very briefly talked about other algorithms that are of interest in Graph theory

Today…

Dynamic Programming

- Forward Approach
- Backward Approach

Example of Dynamic Programming (Hands on)

Fibonacci sequence

Fibonacci sequence: 0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , ...

$F_i = i \qquad \text{if } i \leq 1$

$F_i = F_{i-1} + F_{i-2} \quad \text{if } i \geq 2$

Solved by a recursive program: O(N

Much replicated computation is done.

It should be solved by a simple loop.

Fibonacci sequence

Fibonacci sequence: 0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , …

$F_i = i$ if $i \leq 1$

$F_i = F_{i-1} + F_{i-2}$ if $i \geq 2$

Solved by a recursive program: O(N

Much replicated computation is done.

It should be solved by a simple loop.

Dynamic Programming

Dynamic Programming is an algorithm design method that can be used when the solution to a problem may be viewed as the result of a sequence of decisions

Comparison with divide-and-conquer

Divide-and-conquer algorithms split a problem into separate sub problems, solve the sub problems, and combine the results for a solution to the original problem

- Example: Quicksort
- Example: Mergesort
- Example: Binary search

Divide-and-conquer algorithms can be thought of as top-down algorithms

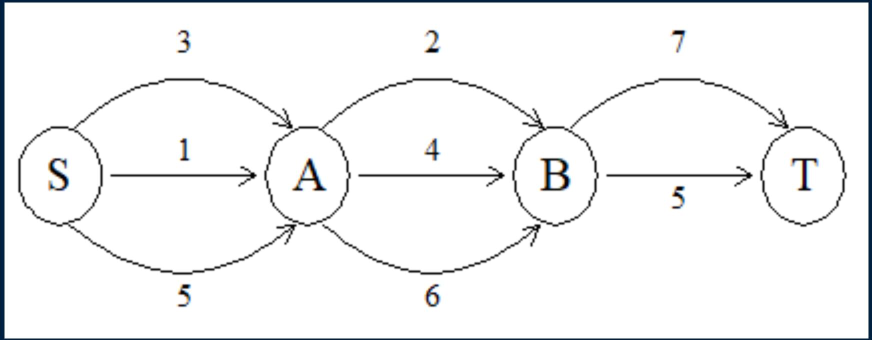
Comparison with divide-and-conquer

In contrast, a dynamic programming algorithm proceeds by solving small problems, remembering the results, then combining them to find the solution to larger problems

Dynamic programming can be thought of as bottom-up

The shortest path

To find a shortest path in a multi-stage graph



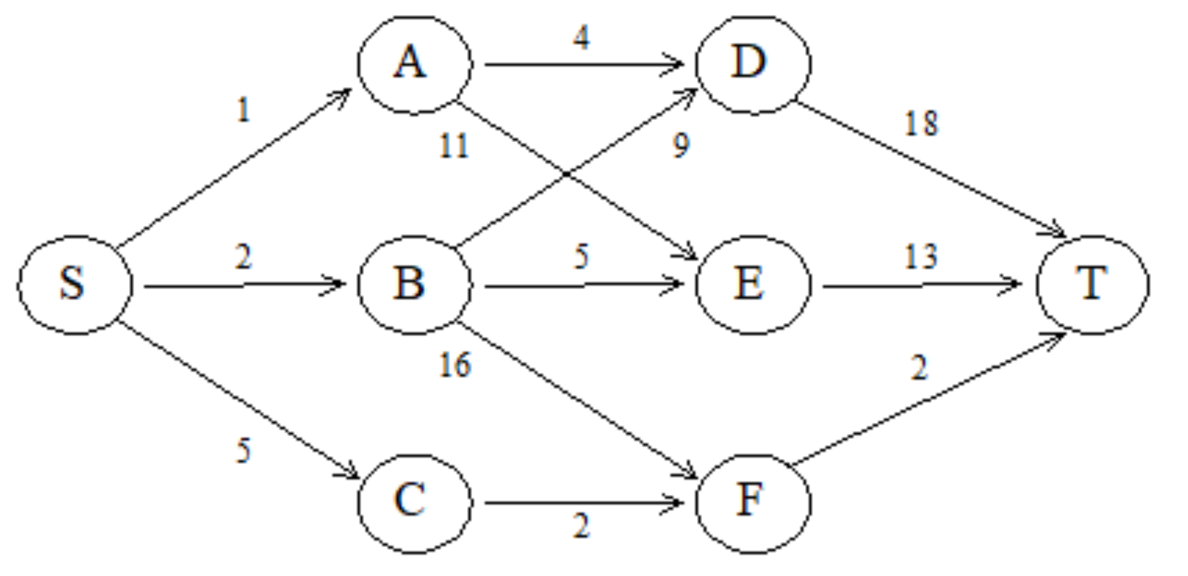
Apply the greedy method :

the shortest path from S to T :

$1 + 2 + 5 = 8$

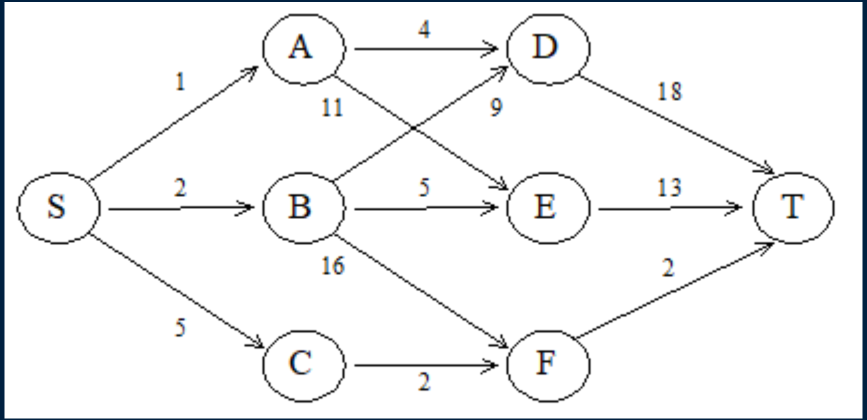
The shortest path in multistage graphs

e.g.



The shortest path in multistage graphs

e.g.



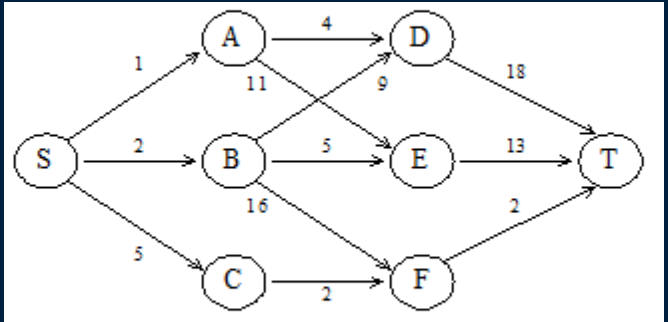
The greedy method can not be applied to this case: (S, A, D, T) 1+4+18 = 23.

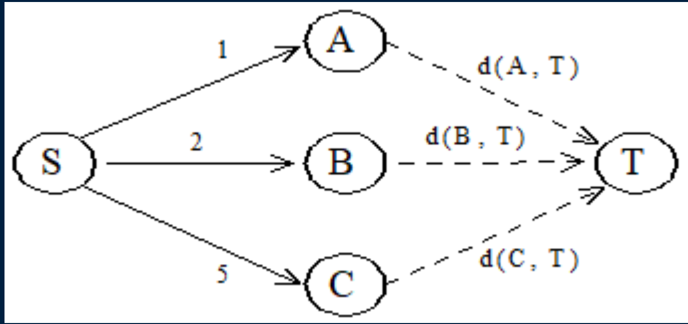
The real shortest path is:

(S, C, F, T) 5+2+2 = 9.

Dynamic programming approach

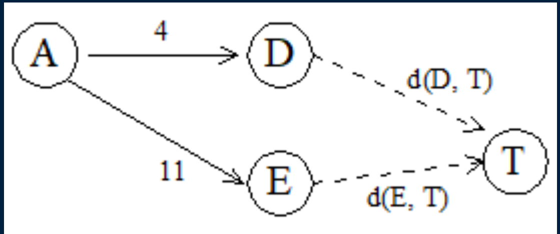
Dynamic programming approach (forward approach):





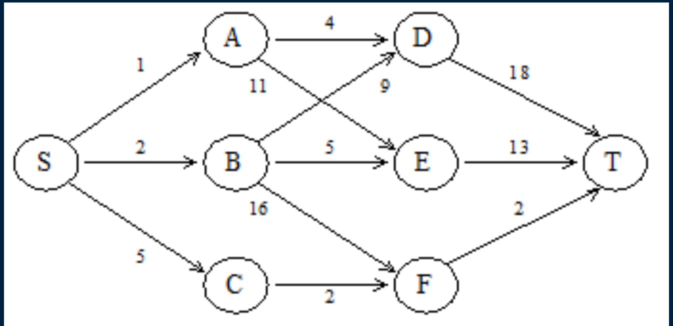
d(S, T) = min{1+d(A, T), 2+d(B, T), 5+d(C, T)}

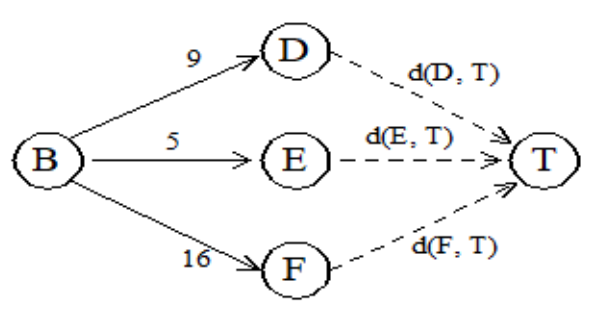
- d(A,T) = min{4+d(D,T), 11+d(E,T)}
  = min{4+18, 11+13} = 22.



$d(B, T) = \min\{9+d(D, T), 5+d(E, T), 16+d(F, T)\}$

$= \min\{9+18, 5+13, 16+2\} = 18.$





$d(C, T) = \min\{ 2+d(F, T) \} = 2+2 = 4$

$d(S, T) = \min\{1+d(A, T), 2+d(B, T), 5+d(C, T)\}$

$= \min\{1+22, 2+18, 5+4\} = 9.$

The above way of reasoning is called backward reasoning.

Backward approach (forward reasoning)

d(S, A) = 1

d(S, B) = 2

d(S, C) = 5

d(S,D)=min{d(S,A)+d(A,D), d(S,B)+d(B,D)}

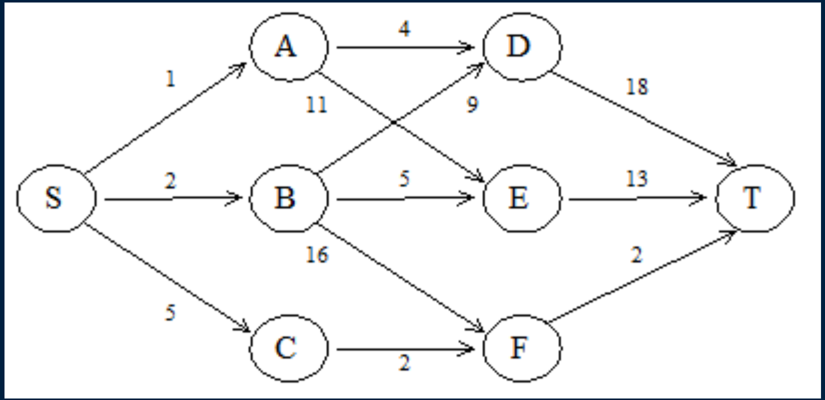
= min{ 1+4, 2+9 } = 5

d(S,E)=min{d(S,A)+d(A,E), d(S,B)+d(B,E)}

= min{ 1+11, 2+5 } = 7

d(S,F)=min{d(S,B)+d(B,F), d(S,C)+d(C,F)}

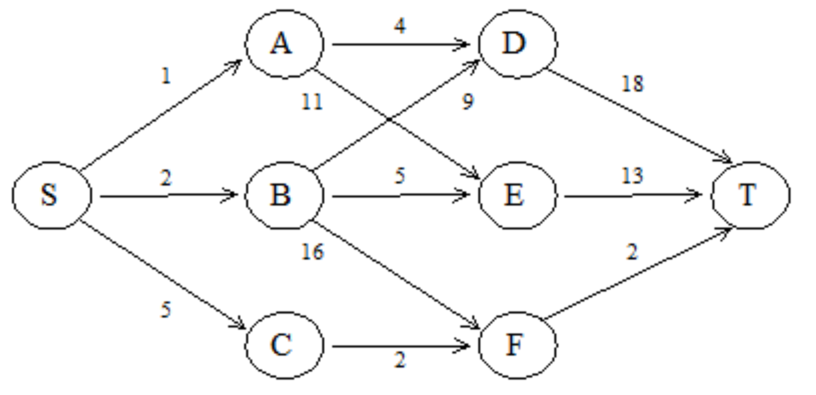
= min{ 2+16, 5+2 } = 7



$$d(S,T) = \min\{d(S, D)+d(D, T),\ d(S,E)+d(E,T),\ d(S, F)+d(F, T)\}$$

$$= \min\{ 5+18,\ 7+13,\ 7+2 \}$$

$$= 9$$



The principle of optimality, I

Dynamic programming is a technique for finding an *optimal* solution

The principle of optimality applies if the optimal solution to a problem can be obtained by combining the optimal solutions to all subproblems.

The principle of optimality, I

Example: Consider the problem of making N¢ with the fewest number of coins

- Either there is an N¢ coin, or
- The set of coins making up an optimal solution for N¢ can be divided into two nonempty subsets, n_1¢ and n_2¢
  - If either subset, n_1¢ or n_2¢, can be made with fewer coins, then clearly N¢ can be made with fewer coins, hence solution was *not* optimal

The principle of optimality, II

The principle of optimality holds if

- Every optimal solution to a problem contains...
- ...optimal solutions to all subproblems

The principle of optimality does *not* say

- If you have optimal solutions to all subproblems...
- ...then you can combine them to get an optimal solution

The principle of optimality, II

Example: In coin problem,

- The optimal solution to 7¢ is 5¢ + 1¢ + 1¢, *and*
- The optimal solution to 6¢ is 5¢ + 1¢, *but*
- The optimal solution to 13¢ is *not* 5¢ + 1¢ + 1¢ + 5¢ + 1¢

But there is *some* way of dividing up 13¢ into subsets with optimal solutions that will give an optimal solution for 13¢

- Hence, the principle of optimality holds for this problem

Example: In coin problem,

- The optimal solution to 7¢ is 5¢ + 1¢ + 1¢, *and*
- The optimal solution to 6¢ is 5¢ + 1¢, *but*
- The optimal solution to 13¢ is *not* 5¢ + 1¢ + 1¢ + 5¢ + 1¢

But there is *some* way of dividing up 13¢ into subsets with optimal solutions that will give an optimal solution for 13¢

- Hence, the principle of optimality holds for this problem

Principle of optimality

Principle of optimality: Suppose that in solving a problem, we have to make a sequence of decisions $D_1, D_2, \ldots, D_n$. If this sequence is optimal, then the last k decisions, $1 < k < n$ must be optimal.

e.g. the shortest path problem

If $i, i_1, i_2, \ldots, j$ is a shortest path from i to j, then $i_1, i_2, \ldots, j$ must be a shortest path from i_1 to j

In summary, if a problem can be described by a multistage graph, then it can be solved by dynamic programming.

Dynamic programming

Forward approach and backward approach:

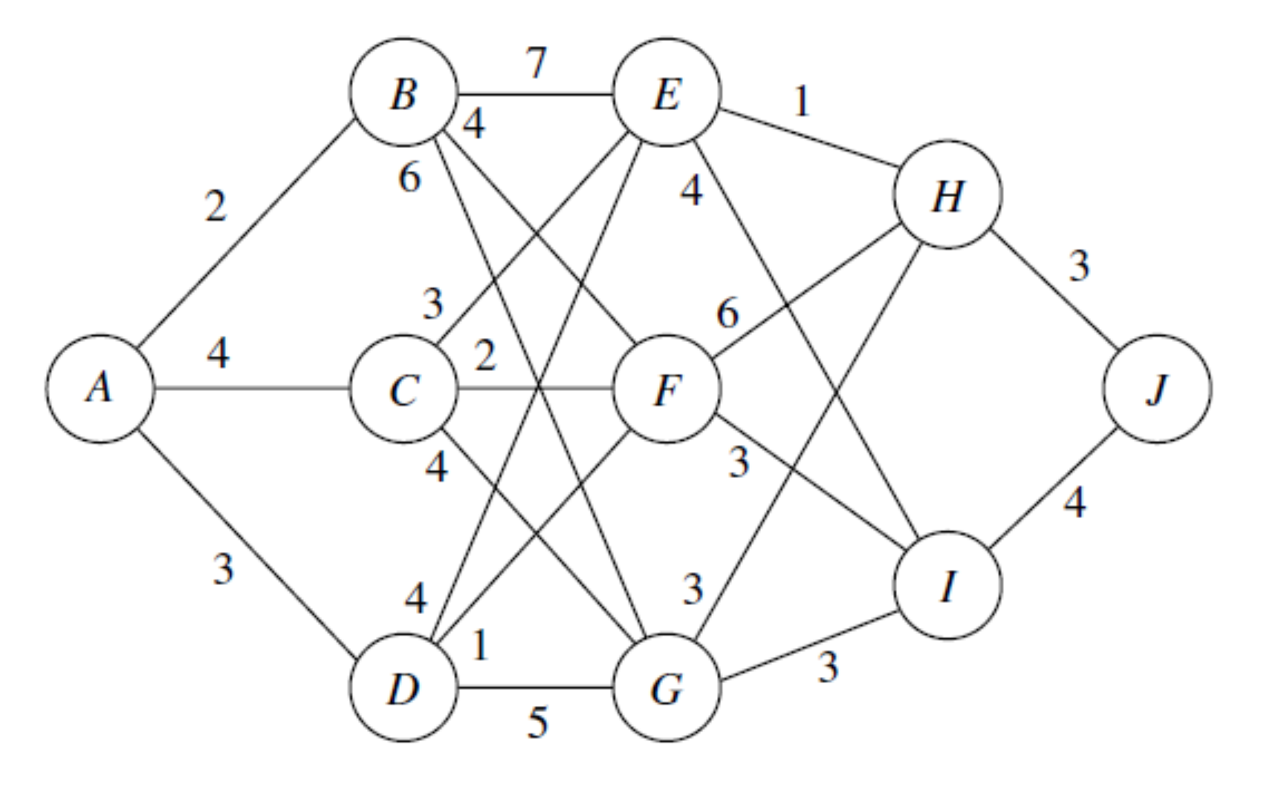
- Note that if the recurrence relations are formulated using the forward approach then the relations are solved backwards . i.e., beginning with the last decision
- On the other hand if the relations are formulated using the backward approach, they are solved forwards.

To solve a problem by using dynamic programming:

- Find out the recurrence relations.
- Represent the problem by a multistage graph.

Hands on Example

Stage Coach Problem – Hands on Example



Recap

We looked introduction to Dynamic Programming

Notion of Forward Approach (Backward Reasoning) and vice-versa

Principle of Optimality in Dynamic Programming

Hands on Example


UBC
THE UNIVERSITY OF BRITISH COLUMBIA