Multistage graph
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 greedy method cannot 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 (forward approach):

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

\[ d(A, T) = \min\{4+d(D, T), 11+d(E, T)\} \]

\[ = \min\{4+18, 11+13\} = 22. \]
Dynamic programming

\[ d(B, T) = \min\{9 + d(D, T), 5 + d(E, T), 16 + d(F, T)\} \]
\[ = \min\{9 + 18, 5 + 13, 16 + 2\} = 18. \]
\[ d(C, T) = \min\{2 + d(F, T)\} = 2 + 2 = 4 \]
\[ d(S, T) = \min\{1 + d(A, T), 2 + d(B, T), 5 + d(C, T)\} \]
\[ = \min\{1 + 22, 2 + 18, 5 + 4\} = 9. \]

The above way of reasoning is called \textit{backward reasoning}.
Backward approach (forward reasoning)

- \( d(S, A) = 1 \)
- \( d(S, B) = 2 \)
- \( d(S, C) = 5 \)

\[
d(S, D) = \min\{d(S, A) + d(A, D), d(S, B) + d(B, D)\}
\]
\[
= \min\{1 + 4, 2 + 9\} = 5
\]

\[
d(S, E) = \min\{d(S, A) + d(A, E), d(S, B) + d(B, E)\}
\]
\[
= \min\{1 + 11, 2 + 5\} = 7
\]

\[
d(S, F) = \min\{d(S, A) + d(A, F), d(S, B) + d(B, F)\}
\]
\[
= \min\{2 + 16, 5 + 2\} = 7
\]
\[ d(S, T) = \min\{d(S, D) + d(D, T), d(S, E) + d(E, T), d(S, F) + d(F, T)\} \]
\[ = \min\{5 + 18, 7 + 13, 7 + 2\} \]
\[ = 9 \]
Principle of optimality

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

e.g. the shortest path problem
If $i, i_1, i_2, ..., j$ is a shortest path from $i$ to $j$, then $i_1, i_2, ..., j$ 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.
The resource allocation problem

- m resources, n projects
- profit $p(i, j)$: j resources are allocated to project i.
- maximize the total profit.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
The multistage graph solution

The resource allocation problem can be described as a multistage graph.

(i, j) : i resources allocated to projects 1, 2, ..., j

E.g. node H=(3, 2) : 3 resources allocated to projects 1, 2.
Find the longest path from S to T:
(S, C, H, L, T),  8+5+0+0=13
2 resources allocated to project 1.
1 resource allocated to project 2.
0 resource allocated to projects 3, 4.
The traveling salesperson (TSP) problem

- e.g. a directed graph:

  ![Graph Image]

  Cost matrix:

  \[
  \begin{pmatrix}
  1 & 2 & 3 & 4 \\
  1 & \infty & 2 & 10 & 5 \\
  2 & 2 & \infty & 9 & \infty \\
  3 & 4 & 3 & \infty & 4 \\
  4 & 6 & 8 & 7 & \infty \\
  \end{pmatrix}
  \]
A multistage graph can describe all possible tours of a directed graph.

Find the shortest path: \((1, 4, 3, 2, 1)\) \(5+7+3+2=17\)