An Introduction to Dynamic Programming

Dynamic programming views the solution to a problem instance as the result of a sequence of solutions to progressively larger sub-instances (starting from the simplest), which eventually lead to the solution to the given instance. *It draws its efficiency from carefully avoiding the requirement of calculating the same quantity more than once.*

- **Divide-and-conquer technique versus dynamic programming**: The former is a top-down approach, whereas the latter is a bottom-up approach.

- **Greedy technique versus dynamic programming**: The former first makes a decision (using a greedy criterion) and then solves the subproblems; the latter first solves the subproblems and then makes a decision.

Dynamic programming relies on a useful principle called the **principle of optimality**, which in many settings appears so natural that it is invoked almost without thinking. *It states that an optimal decision sequence must consist of optimal decision subsequences.*

- **Principle of optimality holds in the case of tracing a shortest path**. Suppose that we trace a shortest path in a graph from node $A$ to node $Y$ as follows: $A \rightarrow W \rightarrow X \rightarrow Y$. Then the portion of this shortest path between any two intermediate nodes (say, $W$ and $Y$) must itself be a shortest path between the respective nodes.

- **Principle of optimality does not hold in the case of tracing a longest path**. To see this, consider the following graph:

```
    A --- B
   /     /
 W     Z
 /     /
X     Y
```

`
Here is a longest path between $A$ and $B$: $A \longrightarrow W \longrightarrow X \longrightarrow Y \longrightarrow Z \longrightarrow B$. Now consider the portion of this path between $W$ and $Y$: $W \longrightarrow X \longrightarrow Y$, and note that it is not a longest path between the two nodes, since there exists a longer path, viz., $W \longrightarrow A \longrightarrow B \longrightarrow Z \longrightarrow Y$. (Tracing a longest path is, in general, NP-hard.)

Dynamic programming cannot be applied if the principle of optimality does not hold with respect to the given problem.

Following are the steps in a dynamic programming solution:

1. Verify that the principle of optimality holds.
2. Set up the dynamic programming recurrence equations.
3. Solve the dynamic programming recurrence equations towards an optimal solution.
4. Perform a trace-back in which the solution itself is constructed.

It is very tempting to write a recursive program to solve the dynamic programming recurrence. However, unless care is taken to avoid re-computing previously computed values, the resulting program will have prohibitive complexity.

The concept of dynamic programming was devised by Richard E. Bellman (1920 - 1984) in 1953. The idea has since proved to be one of the most useful paradigms of algorithm design.

Important References: