DESIGN & ANALYSIS OF ALGORITHMS (3)

techworldthink • March 09, 2022

7. What do you mean by the term Polynomial time reduction?

A transformation of one problem into another which is computable in polynomial time.

In computational complexity theory, a **polynomial-time reduction** is a method for solving one problem using another. One shows that if a hypothetical subroutine solving the second problem exists, then the first problem can be solved by transforming or reducing it to inputs for the second problem and calling the subroutine one or more times. If both the time required to transform the first problem to the second, and the number of times the subroutine is called is polynomial, then the first problem is polynomial-time reducible to the second.

A polynomial-time reduction proves that the first problem is no more difficult than the second one, because whenever an efficient algorithm exists for the second problem, one exists for the first problem as well. By contraposition, if no efficient algorithm exists for the first problem, none exists for the second either. Polynomial-time reductions are frequently used in complexity theory for defining both complexity classes and complete problems for those classes.

8. Define the term Network Flow and illustrate with an example.

Flow Network is a directed graph that is used for modeling material Flow. There are two different vertices; one is a source which produces material at some steady rate, and another one is sink which consumes the content at the same constant speed. The flow of the material at any mark in the system is the rate at which the element moves.

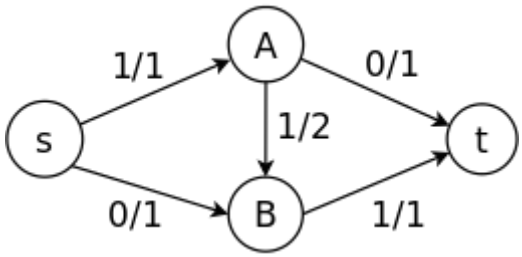
Some real-life problems like the flow of liquids through pipes, the current through wires and delivery of goods can be modeled using flow networks.

Definition: A Flow Network is a directed graph $G = (V, E)$ such that

1. For each edge (u, v) ∈ E, we associate a nonnegative weight capacity c (u, v) ≥ o.If (u, v) ∉ E, we assume that c (u, v) = o.
2. There are two distinguishing points, the source s, and the sink t;
3. For every vertex v ∈ V, there is a path from s to t containing v.

Let G = (V, E) be a flow network. Let s be the source of the network, and let t be the sink. A flow in G is a real-valued function f: V x V→R such that the following properties hold:

- Capacity Constraint: For all u, v ∈ V, we need f (u, v) ≤ c (u, v).
- Skew Symmetry: For all u, v ∈ V, we need f (u, v) = - f (u, v).
- Flow Conservation: For all u ∈ V-{s, t}, we need



Flow network showing capacities and flows. An edge labeled x / yx/y has flow xx and capacity yy

The quantity f (u, v), which can be positive or negative, is known as the net flow from vertex u to vertex v. In the maximum-flow problem, we are given a flow network G with source s and sink t, and we wish to find a flow of maximum value from s to t.

The three properties can be described as follows:

1. Capacity Constraint makes sure that the flow through each edge is not greater than the capacity.
2. Skew Symmetry means that the flow from u to v is the negative of the flow from v to u.
3. The flow-conservation property says that the total net flow out of a vertex other than the source or sink is o. In other words, the amount of flow into a v is the same as the amount of flow out of v for every vertex v ∈ V - {s, t}

9. What do you mean by approximation ratio of an Approximation algorithm?

The **approximation ratio** (or **approximation factor**) of an algorithm is the ratio between the result obtained by the algorithm and the optimal cost or profit. Typically this ratio is taken in whichever direction makes it bigger than one;

> *for example, an algorithm that solves for a cost of $2 an instance of a problem that has an optimal cost of $1 has approximation ratio 2;but an algorithm that sells 10 airplane tickets (a profit of 10) when the optimum is 20 also has approximation ratio 2.*

An algorithm with approximation ratio k is called a k-approximation algorithm; both algorithms above would be called 2-approximation algorithms.

When the approximation ratio is close to 1, it is often more useful to look at the **approximation error**, which is defined as the approximation ratio minus 1. So an algorithm that always got within 1.01 of the optimal cost or profit would have a 1% approximation error.

An Approximate Algorithm is a way of approach NP-COMPLETENESS for the optimization problem. This technique does not guarantee the best solution. The goal of an approximation algorithm is to come as close as possible to the optimum value in a reasonable amount of time which is at the most polynomial time. Such algorithms are called approximation algorithm or heuristic algorithm.

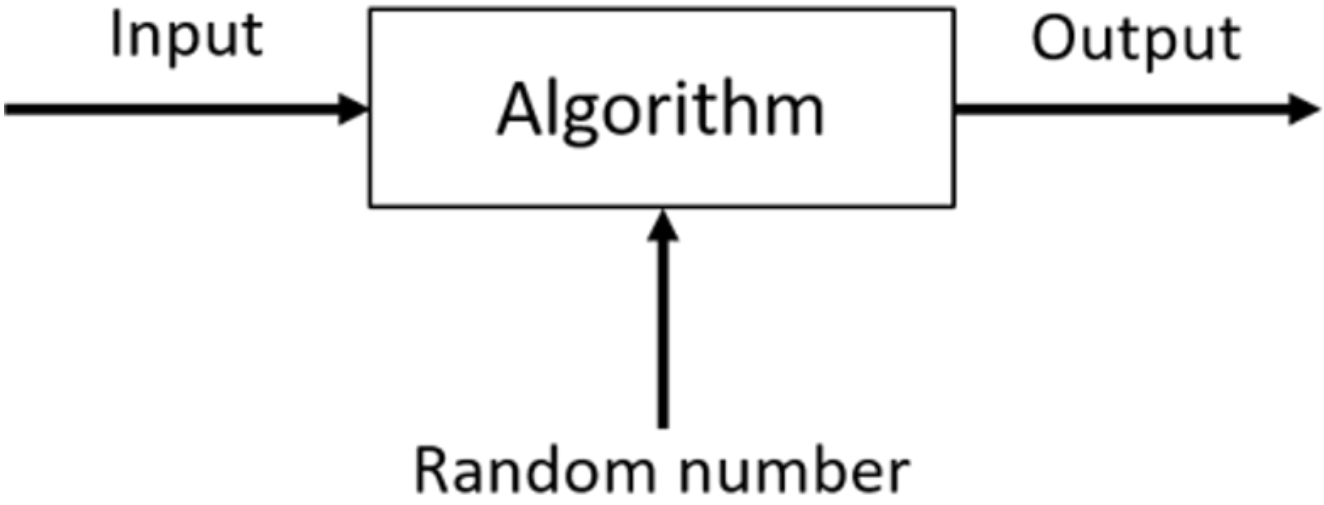
- For the traveling salesperson problem, the optimization problem is to find the shortest cycle, and the approximation problem is to find a short cycle.
- For the vertex cover problem, the optimization problem is to find the vertex cover with fewest vertices, and the approximation problem is to find the vertex cover with few vertices.

10. What is meant by a Randomised Algorithm?

A randomized algorithm is defined as an algorithm that is allowed to access a source of independent, unbiased random bits, and it is then allowed to use these random

bits to influence its computation.A randomized algorithm uses a random number at least once during the computation make a decision.

Randomized Algorithm refers to an algorithm that uses random numbers to determine what to do next at any point in its logic. In a standard algorithm, it is usually used to reduce either the running time, or time complexity, or the memory used, or space complexity. The algorithm works by creating a random number, r, from a set of numbers and making decisions based on its value. This algorithm could assist in making a decision in a situation of doubt by flipping a coin or drawing a card from a deck.



Randomized Algorithm

Randomized Algorithm Flowchart

When utilizing a randomized method, keep the following two considerations in mind:

- It takes source of random numbers and makes random choices during execution along with the input.
- Behavior of an algorithm varies even on fixed inputs.

For example, in Randomized Quick Sort, we use a random number to pick the next pivot (or we randomly shuffle the array). And in Karger's algorithm, we randomly pick an edge.