Algorithm Design Manual Solutions

Zachary William Grimm

Solutions to Selected Problems

zwgrimm@gmail.com

2/2/2019

1 Introduction To Algorithm Design

Finding Counter Examples

1-1. *Show that a + b can be less than min(a, b)*

$$\begin{gathered}\text{Let } a=-1, b=-1 \\ \text{Then } a+b=-2,\ min(a,b)=-1 \\ \therefore \exists\ a,b \in Z : a+b < min(a,b)\end{gathered}$$

1-2. *Show that a * b can be less than min(a, b)*

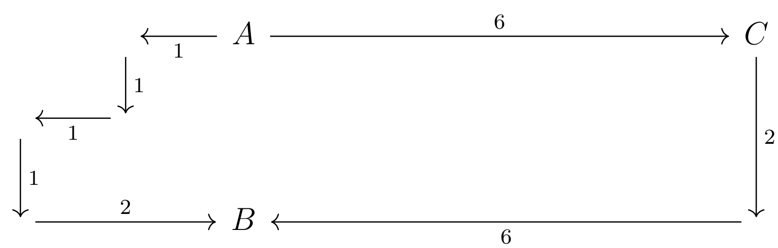
$$\begin{gathered}\text{Let } a=-1, b=5. \\ \text{Then } a*b=-5,\ min(a,b)=-1 \\ \therefore \exists\ a,b \in Z : a*b < min(a,b)\end{gathered}$$

1-3. *Design/draw a road network with two points a and b such that the fastest route between a and b is not the shortest route*

$$\begin{tikzcd}
A \arrow[r, "D{=}5m,\ S{=}1m/s"] \arrow[dr, "D{=}6m,\ S{=}3m/s"'] & C \arrow[r, "D{=}5m,\ S{=}.2m/s"] & B \\
& D \arrow[ur, "D{=}6m,\ S{=}3m/s"'] &
\end{tikzcd}$$

Although the distance from A to B through C is shorter than going through D, road constraints limit the time it takes making the route through D faster despite it being longer.

1-4. *Design/draw a road network with two points a and b such that the shortest route between a and b is not the route with the fewest turns*



The route from A through C to B has only two turns but is a total length of 14 units while the direct route from A to B (the shortest) has 4 turns and is a length of 6 units. $\therefore$ The shortest route between A and B is not the route with the fewest turns.

1-5. ***The knapsack problem is as follows:*** *Given a set of integers S = s1, s2,...,sn, and a target number T, find a subset of S which adds up exactly to T. For example, there exists a subset within S = 1, 2, 5, 9, 10 that adds up to T = 22 but not T = 23. Find counterexamples to each of the following algorithms for the knapsack problem. That is, giving an S and T such that the subset is selected using the algorithm does not leave the knapsack completely full, even though such a solution exists.*

(a) ***Put the elements of S in the knapsack in left to right order if they fit, i.e. the first-fit algorithm.***
Let $S = \{1, 7, 9\},\ T = 10$

(b) ***Put the elements of S in the knapsack from smallest to largest, i.e. the best-fit algorithm.***
Let $S = \{1, 7, 9\},\ T = 10$

(c) ***Put the elements of S in the knapsack from largest to smallest.***
Let $S = \{1, 4, 5, 7, 9\},\ T = 19$