Henzi Kou

CIS 315: Intermediate Algorithms

Christopher Wilson

4 June 2019

Assignment 7

1. **(a)** Give an efficient (greedy!) algorithm for computing the optimal order in which to process the jobs so that the completion time is minimized.

A greedy method to approach this problem is to take and process the jobs with longest finishing time to the smallest finishing time; in decreasing order. In order to accomplish this, we must sort the run times of jobs *fi* in descending order, e.g. , which will take .

**(b)** Describe the greedy choice your algorithm makes and show that it is correct.

In order to prove this algorithm holds, we will prove it through contradiction. Suppose that the case stated above is *not* true such that there exists a case *i* so that *fi* < *fi*+1. Since the previous statement must be true, then we have to swap the positions of *Ji* and *Ji*+1 because *Ji*+1 will have a shorter completion time than *Ji*. Given *Ji*, we know that the completion time, *t*, for it will be *ti* = *s* + *si* + *fi*, where *s* is the start time of *Ji*. Similarly, we have the completion time for *Ji*+1 is *ti*+1 = *s* + *si* + *si*+1 + *fi*+1. By swapping positions of *Ji* and *Ji*+1 we have *Ji*+1 at position *i*, we can denote the new change in position of completion time as *t’i*+1. Now we must prove that the completion times of *t’i*, *t’i*+1 is less than *ti*, *ti*+1. Thus, we have the following,

Therefore, we have shown that the *t’i*, *t’i*+1 is less than *ti*, *ti*+1.

1. Exercise 5.18, part (a), left column only (DPV). What is the optimum Huffman encoding of this alphabet?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **h** | **s** | **n** | **i** | **o** | **a** | **t** | **e** | **blank** |
| 4.9 | 5.1 | 5.5 | 5.8 | 5.9 | 6.8 | 7.7 | 10.2 | 18.3 |

(Intermediate steps shown next page.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **h** | **s** | **n** | **i** | **o** | **a** | **t** | **e** | **blank** |
| 0110 | 0111 | 1110 | 1111 | 000 | 001 | 010 | 110 | 10 |
| 4 bits | 4 bits | 4 bits | 4 bits | 3 bits | 3 bits | 3 bits | 3 bits | 2 bits |

1. Illustrate the Ford-Fulkerson algorithm on the graph of Figure 1.