

1. [10 pt] Show by induction that mergesort is $\Theta(n \lg n)$.
2. [10 pt] Use the master's method to get a solution to the following recurrences:
 - (a) $T_n = 4T_{n/2} + n$

(a) $T_n = 4T_{n/2} + n$

(b) $T_n = cT_{n-i} + in$, where c and i are some constants

3. [10 pt] A version of binary search algorithm divides the array into two parts that are not equal but one part is twice the size of the other part. For convenience, you can assume that the smaller part is the first part. Write a recurrence corresponding to this search and solve it using any method of your choice.

4. [10 pt] Find an optimal solution to the 0/1 knapsack instance when $n = 7$, $m = 15$, $(p_1, p_2, \dots, p_7) = (10, 5, 15, 7, 6, 18, 3)$, and $(w_1, w_2, \dots, w_7) = (2, 3, 5, 7, 1, 4, 1)$.

5. [10 pt] Consider the following variables in the assembly line scheduling problem:

- (a) $l_i[j]$ – Line number whose station $j - 1$ is used in a fastest way through station $S_{i,j}$. $l_i[1]$ is undefined.
- (b) l^* – Line whose station n is used in a fastest way through the entire factory

The above information is used to print the stations used, in decreasing order of station number, in the following algorithm:

```
print_stations ( l, n )
{
    i = l*;          // i is a pointer to a node in line l*
    print "line ", i, " station ", n
    for ( j = n-1; j > 1; j-- )
    {
        i = l_i[j]
        print "line ", i, " station ", j-1
    }
}
```

An example output is:

```
line 1, station 6
line 2, station 5
line 2, station 4
line 1, station 3
line 2, station 2
line 1, station 1
```

Modify this algorithm to print the stations in increasing order of station number.