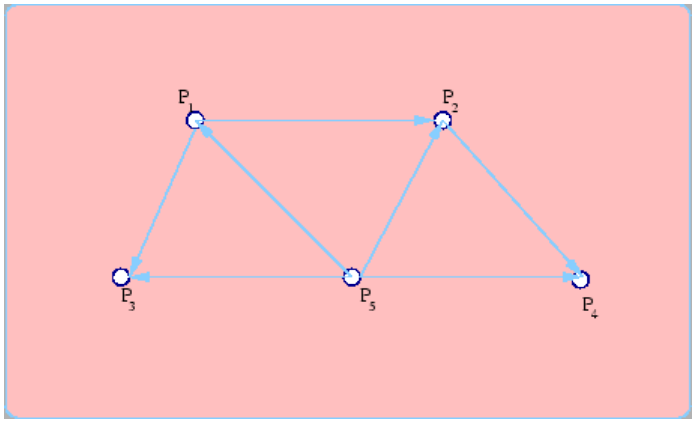
Design and Analysis of Algorithm

Practice Set

Note: It is just a practice set. You will find the similar questions, not exact ones. Therefore, practice as much as you can.

1. What is Asymptotic Notation?
2. Apply the Master Theorem, Substitution method, and recursive tree method in the following
    a. T(n) = 9T(n/3) + n^3
    b. T(n) = 4T(n/2) + n
3. Write down the limitations of the master theorem and substitution method.
4. Write iterative and recursive pseudocode of linear search, binary search, sum of n elements, Fibonacci series, and factorial of a number
5. Explain all asymptotic properties with an example.
6. Analyze the best case and worst case complexities of merge sort, quick sort, and heap sort
7. Describe Strassen's Matrix multiplication and explain how complexity is reduced to O(n^2.8)
8. What is a Minimum Spanning Tree (MST)? Name two greedy approaches that are used to solve MST.
9. Explain Prim's and Kruskal's algorithms, mentioning their pseudocode and complexity.
10. What is the purpose of using sorted edges and the disjoint set data structure in Kruskal's algorithm?
11. Explain the three main approaches to string matching in detail— Naïve, Rabin-Karp, and Knuth-Morris-Pratt (KMP) — including their algorithms, pseudocode, and time complexity analysis.
12. Apply Naïve, Rabin-Karp, and KMP methods of pattern matching in the following Text: "THIS IS A PRACTICE SET", Pattern: "CTICE"
13. Find the minimum cost using Prim's and Kruskal's algorithms in the following graph and adjacency matrix. Consider P1 as Source Vertex.
    a. 

    b.

	P_1	P_2	P_3	P_4	P_5
P_1	0	1	1	0	1
P_2	0	0	0	1	1
P_3	0	0	0	0	0
P_4	0	0	0	0	0
P_5	0	0	1	1	0

14. Find the complexity of the following non-recursive algorithms:

a.

```
for i from 0 to n-1:
    if A[i] == key:
        return i
return -1
```

b.

```
for i from 1 to n-1:
    key = A[i]
    j = i - 1
    while j >= 0 and A[j] > key:
        A[j + 1] = A[j]
        j = j - 1
    A[j + 1] = key
```

c.

Analysis Ex.

```
main()
{
    while(n>2)
    {
        n = n^(1/2);
    }
}
```

d.

Analysis Ex.

```
main()
{
    i=1;
    while(i<n)
    {
        i= 2*i;
    }
}
```

15. Write the pseudocode for Quick Sort, Heap Sort, and Merge Sort algorithms. Also, discuss their advantages and disadvantages.
16. Generate the recurrence relation of quick sort, merge sort, binary search, Fibonacci sequence, Tower of Hanoi, Strassen's Matrix Multiplication, and find their complexity.