

系別：資訊工程學系 A 組

科目：資料結構

考試日期：3 月 4 日(星期六) 第 2 節

本試題共 七 大題， 2 頁

本試題雙面印刷

一、Please illustrate the Big Oh of the following function, it means what is its $O(g(n))$ and why? (10%)

```
void Matrix_Mul(&C, &A, &B) {
    for (i=1; i<=m; i++) {
        for (j=1; j<=p; j++) {
            C[i][j]=0; /*initializing the element of C array */
            for(k = 1; k <= n; k++) {
                C[i][j] = C[i][j] + A[i][k] * B[k][j]; /*calculation of the element */
            }
            printf("%d ", C[i][j]); /*finish one element */
        }
        printf("\n");
    }
}
return;
```

二、The original sequence of integer number is following, 99, 48, 11, 5, 77, 18, 70, 55, 0, 3, please illustrate how to sorting them in increasing order by the three sorting method, respectively. (30%, 10% each)

- (1) quick sort
- (2) heap sort
- (3) bubble sort

三、An array with size is $A[8][7]$, and each element is occupied 32bits. The memory address uses Byte as assigned unit. Now, assume $A[0][0]$ is allocated at 1024, please find the location (address) of $A[0][7]$ and $A[6][5]$? Please give your answer using row-major and column-major, respectively. (20%, 5% each)

四、An arithmetic expression in infix representation is following,
 $X = A + B * C - D / E * F ^ I ^ G + H * J / K$, which $^$ is exponent operator
Please give the prefix and postfix representations, respectively. (10%, 5% each)

淡江大學 106 學年度碩士班招生考試試題

43-2

系別：資訊工程學系 A 組

科目：資料結構

考試日期：3 月 4 日(星期六) 第 2 節

本試題共 七 大題， 2 頁

五、Following the figure 1, please draw its Adjacency List, DFS, and BFS spanning trees started from Vertex 9, respectively. Here the Adjacency List is linked with increasing ordering and the Graph traversal is traced by the specific Adjacency List. (4%, 3%, 3%)

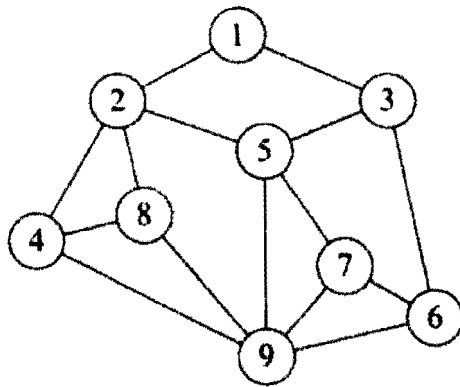


Figure 1.

六、Following the figure 2, please draw its minimal cost spanning tree generated by Kruskal's algorithm and Prim's algorithm, and what is the value of minimal cost, respectively? (5%, 5%)

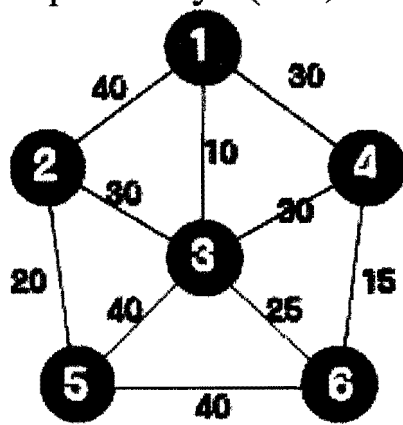


Figure 2.

七、Following the figure 3, please find the shortest paths from Vertex "0" to other vertices followed the Dijkstra algorithm. (10%)

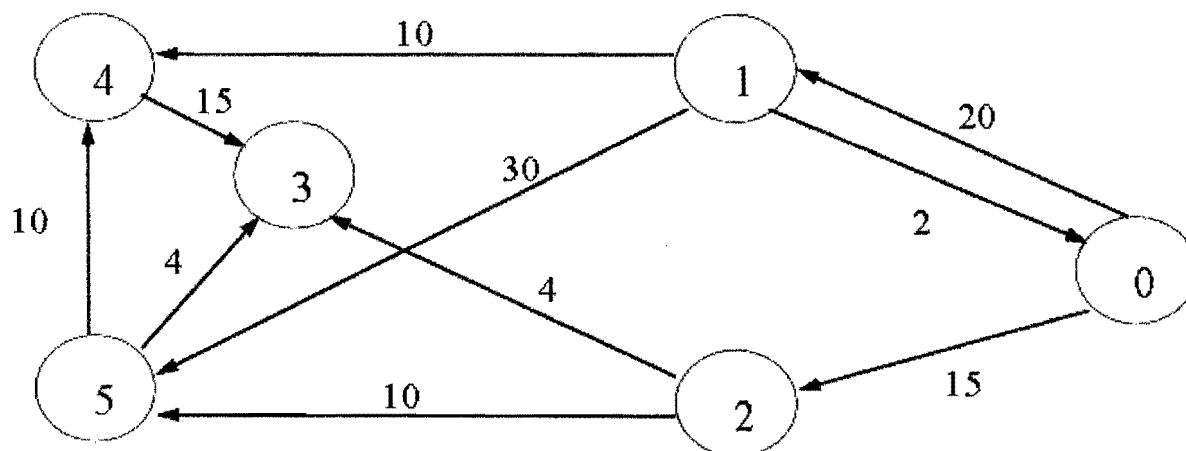


Figure 3.