

1. จงเขียนโปรแกรมเพื่อหาค่าสูงสุดหรือต่ำสุดในลิสต์ n ค่าจาก N ค่าได้อย่างไรโดยใช้เวลาน้อยที่สุด คิดเหมือน QuickSort คำตอบไม่จำเป็นต้องเรียงกันก็ได้ จะใช้วิธีไหนก็ได้ ให้คะแนนจาก time complexity

อธิบาย code

ใช้ sort linked list ตัวเลขทั้งหมด N จำนวน ต้องการค่ามากที่สุดและน้อยสุด n จำนวน โดยแบ่งเป็น 2 ฟังก์ชัน

1. less ใช้หาค่าน้อยสุด n จำนวน โดยจะมี Big $O(n*N)$

นำข้อมูลทั้งหมด(N)มา sort โดยเช็คตามลำดับ $O(N)$

วิธีการ sort นำเลขมาเช็คตามลำดับ n ครั้ง $O(n*N)$

2. greater ใช้หาค่ามากที่สุด n จำนวน โดยวิธีการทำงานเหมือนฟังก์ชันที่ 1

จะได้ Big $O(2*n*N)$ เมื่อต้องการหาทั้งค่ามากและค่าน้อย n จำนวน

$$\text{Big Omega} = O(2*N)$$

time complexity = $O(N)$

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4
5 class Node
6 {
7     public:
8         int num;
9         Node *next;
10 };
11
12 class arrange
13 {
14     public:
15         Node *head;
16         Node *fence;
17
18     public:
19         arrange()
20         {
21             head = NULL;
22         }
23
24         void less(int n, int num)
25         {
26             Node* newnode = new Node;
27             newnode->num = num;
28             newnode->next = NULL;
29
30             if(head == NULL)
31             {
32                 head = newnode;
33             }
34             else
35             {
36                 if(num < head->num)
37                 {
38                     newnode->next = head;
39                     head = newnode;
40                 }
41                 else
42                 {
43                     fence = head;
44                     for(int i = 0; i < n; i++)
45                     {
```

```
46         if(fence->next == NULL)
47         {
48             fence->next = newnode;
49             return;
50         }
51         else if(num < fence->next->num)
52         {
53             newnode->next = fence->next;
54             fence->next = newnode;
55             return ;
56         }
57         fence = fence->next;
58     }
59 }
60 }
61 }
62
63 void greater(int n, int num)
64 {
65     Node* newnode = new Node;
66     newnode->num = num;
67     newnode->next = NULL;
68
69     if(head == NULL)
70     {
71         head = newnode;
72     }
73     else
74     {
75         if(num > head->num)
76         {
77             newnode->next = head;
78             head = newnode;
79         }
80         else
81         {
82             fence = head;
83             for(int i = 0; i < n; i++)
84             {
85                 if(fence->next == NULL)
86                 {
87                     fence->next = newnode;
88                     return;
89                 }
90                 else if(num > fence->next->num)
91                 {
```

```
92         newnode->next = fence->next;
93         fence->next = newnode;
94         return ;
95     }
96     fence = fence->next;
97 }
98 }
99 }
100 }
101
102 void show(int n)
103 {
104     fence = head;
105     for(int j = 0; j < n; j++)
106     {
107         cout << fence->num << " ";
108         fence = fence->next;
109     }
110     cout << endl;
111 }
112 };
113
114
115 int main()
116 {
117     arrange data;
118     int lst[] = {852,405,849,1000,324,258,267,214,900,
119                 496,186,599,535,2,150,763,336,121,553,
120                 829,584,506,621,375,490,727,101,98,731,
121                 682,624,810,389,782,400,351,688,833,429,
122                 879,819,226,820,215,828,249,614,441,171,
123                 201,860,796,568,379,899,844,9,1203};
124     int n;
125     int N = sizeof(lst) / sizeof(lst[0]);
126
127     cout << "\nlist numbers : ";
128     for(int i = 0; i < N; i++)
129     {
130         cout << lst[i] << " ";
131     }
132     cout << endl << endl;
133
134     cout << "How many numbers : ";
135     cin >> n;
136
137     for(int i = 0; i < N; i++)
```

```
138     {
139         data.less(n, lst[i]);
140     }
141     cout << "\n" << n << " numbers sorted from least\n";
142     data.show(n);
143
144     for(int i = 0; i < N; i++)
145     {
146         data.greater(n, lst[i]);
147     }
148     cout << "\n" << n << " numbers sorted from highest\n";
149     data.show(n);
150
151     cout << endl;
152 }
```

OUTPUT

```
list numbers : 852 405 849 1000 324 258 267 214 900 496 186 599 535
2 150 763 336 121 553 829 584 506 621 375 490 727 101 98 731 682 624
810 389 782 400 351 688 833 429 879 819 226 820 215 828 249 614 441
171 201 860 796 568 379 899 844 9 1203
```

How many numbers : 10

```
10 numbers sorted from least
2 9 98 101 121 150 171 186 201 214
```

```
10 numbers sorted from highest
1203 1000 900 899 879 860 852 849 844 833
```