

هر سوال را در محل در نظر گرفته شده پاسخ دهید. پاسخ های خارج از محل تصحیح نمیشوند.

۱. [۲۵] پیچیدگی محاسباتی $O(n^2)$ و حافظه ای $O(n)$ متد $G(n)$ چیست؟ دلیل خود برای هر دو مورد را بطور مختصر در نقطه چین زیر توضیح دهید.

```
1 public static int G(int n)
2 {
3     int sum = 0;
4     // Assumption: LinkedList is a singly linked list data structure.
5     // It also maintains a pointer to the last element.
6     LinkedList<int> s = new LinkedList<int>();
7     for(int i=0; i<n; i++)
8         s.AddLast(n-i);
9
10    while(s.Count > 0)
11    {
12        sum += s.Last();
13        s.RemoveLast();
14    }
15
16    return sum;
17 }
```

پیچیدگی محاسباتی خط هشتم $O(1)$ است (اضافه کردن به لینکد لیست $O(1)$ است) که در یک حلقه n تایی صدا زده میشود. پس پیچیدگی محاسباتی خطوط ۷ و ۸ میشود: $O(n)$
خطوط دهم تا سیزدهم n بار اجرا میشوند که چون پیچیدگی محاسباتی حذف کردن آخرین عنصر از لیست یک طرفه $O(n)$ است پس پیچیدگی محاسباتی این خطوط هم میشود $O(n^2)$ و پیچیدگی محاسباتی متد G در مجموع میشود $O(n^2)$.

2. [50] Assume that the disjoint sets data structure is implemented as an array `smallest[1...12]`: `smallest[i]` is equal to the smallest element in the set containing `i`.

(a) [20] What is the output of the following program? As an answer, enter four integers separated by spaces: 1 3 3 1

```
1 for i from 1 to 12:
2     MakeSet(i)
3 Union(2, 10)
4 Union(7, 5)
5 Union(6, 1)
6 Union(3, 4)
7 Union(5, 11)
8 Union(7, 8)
9 Union(7, 3)
10 Union(12, 2)
11 Union(9, 6)
12 print(Find(6))
13 print(Find(3))
14 print(Find(11))
15 print(Find(9))
```

(b) [20] Assume that the disjoint sets data structure is implemented as disjoint trees with union by rank heuristic. List the height of all trees in the forest separated by space from shortest to tallest: 1 1 2 3

- (c) [10] Assume that the disjoint sets data structure is implemented as disjoint trees with union by rank heuristic and with path compression heuristic. Compute the maximum height of a tree in the resulting forest. (Recall that the height of a tree is the number of edges on a longest path from the root to a leaf. In particular, the height of a tree consisting of just one node is equal to 0.): 1

```

1  for i from 1 to 60:
2      MakeSet(i)
3  for i from 1 to 30:
4      Union(i, 2*i)
5  for i from 1 to 20:
6      Union(i, 3*i)
7  for i from 1 to 12:
8      Union(i, 5*i)
9  for i from 1 to 60:
10     Find(i)

```

۳. [۲۵] می‌خواهیم نمرات دانشجویان زیر را بر اساس شماره دانشجویی در یک HashTable ذخیره کنیم. اندازه آرایه ثابت و برابر ۱۰ می‌باشد. تابع Hash جمع دو شماره آخر شماره دانشجویی می‌شود. مثلا برای ۹۷۵۲۱۱۲۱۲ می‌شود ۳. ساختمان داده‌ای HashTable به همراه محتویات آن را پس از اضافه کردن مقادیر زیر نمایش دهید.

(97522202, 19) (97521486, 20) (97522004, 18) (97522229, 13)
 (96462014, 12) (97522031, 9) (97522049, 15) (97522238, 20)
 (97522067, 12) (96431335, 18) (97522265, 16) (97522085, 17)
 (97522094, 18)

Key	Value	Hash	% m
97522202	19	2	2
97521486	20	14	4
97522004	18	4	4
97522229	13	11	1
96462014	12	5	5
97522031	9	4	4
97522049	15	13	3
97522238	20	11	1
97522067	12	13	3
96431335	18	8	8
97522265	16	11	1
97522085	17	13	3
97522094	18	13	3

0	
1	→ [97522229 13] → [97522238 20] → [97522265 16]
2	→ [97522202 19]
3	→ [97522049 15] → [97522067 12] → [97522085 17] → [97522094 18]
4	→ [97521486 20] → [97522004 18] → [97522031 9]
5	
6	→ [96462014 12]
7	
8	→ [96431335 18]
9	