INF102 FALL 2018

MANDATORY ASSIGNMENT 0

EXAMPLE SOLUTION

TASK 2: BIG-O QUIZ

	f(n)	alt.	~	Big-O
A	$2n + 1$		$2n$	n
B	$2n$		$2n$	n
C	$\frac{n^2}{2} - \frac{n}{2}$		$\frac{n^2}{2}$	n^2
D	$\frac{n^2}{2} + \frac{n}{2} + 2$		$\frac{n^2}{2}$	n^2
E	1		1	1
F	$\left\lceil \frac{n}{2} \right\rceil$		$\frac{n}{2}$	n
G	$2n + 2$		$2n$	n
H	$\lfloor \lg n \rfloor + 2$		$\lg n$	$\log n$
I	$\sum_{i=0}^{\lfloor \lg n \rfloor} \lfloor \frac{n}{2^i} \rfloor$		$2n$	n
J	$\sum_{i=0}^{\lfloor \lg n \rfloor} 2^i \left\lfloor \frac{n}{2^i} \right\rfloor$		$n \lg n$	$n \log n$
K	$\lceil \sqrt{n} \rceil$		$\sqrt{n}$	$\sqrt{n}$
L	$\lfloor \lg n \rfloor + 1$		$\lg n$	$\log n$
M	$\lfloor e \cdot n! \rfloor$	$\sum_{i=0}^{n} \frac{n!}{i!}$	$e \cdot n!$	$n!$
N	$\lceil \lg n \rceil$		$\lg n$	$\log n$
O	$\frac{3^{n+1}}{2} - \frac{1}{2}$	$\sum_{i=0}^{n} 3^i$	$\frac{3^{n+1}}{2}$	3^n
P	$\lfloor \lg n \rfloor \cdot n + 2n$		$n \lg n$	$n \log n$
Q	$\lfloor \lg n \rfloor + 1$		$\lg n$	$\log n$
R	$n + 2\lceil \sqrt{n} \rceil + 1$		n	n

S	$\left\lceil \frac{n^3}{\lceil\sqrt{n}\rceil} \right\rceil \cdot (1 + \lceil\sqrt{n}\rceil)$		n^3	n^3
T	$2n^2\lfloor\sqrt{n-1}\rfloor - \lfloor\sqrt{n-1}\rfloor^2 - \lfloor\sqrt{n-1}\rfloor +$ $n\lfloor\lg n\rfloor - 2^{\lfloor\lg n\rfloor+1} + \lfloor\lg n\rfloor + 3n^2 + 2$		$2n^{2.5}$	$n^{2.5}$
U	$2 + \lfloor\lg n\rfloor + \sum_{i=0}^{\lfloor\lg n\rfloor} \left(\left\lceil \ln \left\lfloor \frac{n}{2^i} \right\rfloor \right\rceil - 1\right)$		$\frac{\ln n \ln 2n}{\ln 4}$	$(\log n)^2$
V	$1 + \sum_{i=0}^{\lfloor\lg n\rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor$		$2n$	n
W	$2n + \lfloor\lg n\rfloor + 1$		$2n$	n
X	$2^{\lfloor\lg n\rfloor+1} + 2\lfloor\lg n\rfloor + 2$		$2n$	n
Y	$\left\lfloor \frac{1+\sqrt{n}}{2} \right\rfloor$		$\frac{\sqrt{n}}{2}$	$\sqrt{n}$
Z*	$2n - 2 + \sum_{i=1}^{\pi(\lfloor\sqrt{n}\rfloor)} \left(\left\lfloor \frac{n}{p_i} \right\rfloor + 1 - p_i \right)$			$n \log\log n$

Thanks to students ono008 (task T) and kkn015 (task Z) for providing better answers than us for some of the exact functions. You have both been awarded a bonus point towards your final grade in the course.

* In task Z, π is the prime counting function (how many primes <= x), and p_i is the i'th prime.

TASK 3: UNION FIND

c) Yes, it is indeed possible. Use weighted quick union as usual, but also keep another array, int[] smallest, which holds the minimum element contained in the corresponding connected component.

d) We start by making calls to union(p, q) with the objective that the elements form a single, long chain. We notice that the ancestor of q will be made the common ancestor; hence, to increase the length of the chain, p must already be in the longest chain, whereas q could be a single element.

We should always pick the element p to be the element at the very end of the longest chain. This will be the element first added to the chain; say element 0 (could be anything, as long as we are consistently choosing the same element as p).

This leads us to the following strategy (assuming m >> n)

```
for (int i=1; i<n; i++) {
      union(0, i)
```

After these calls all have been made, the chain has the maximum possible length (n-1), and element 0 is at the very end of the chain. To truly maximize the number of steps preformed for every function call, we can now call `union(0, 0)` repeatedly until we have reached our allotted number of calls m.

If m < c*n, for some constant c, things become slightly more complicated, since you need to carefully decide when to start making the `union(0, 0)` calls (possibly before making the chain into maximum length). It is left as an exercise for those intrigued to figure out at which exact point it is optimal to switch from `union(0, i)` to `union(0, 0)`. This should also reveal the value c.