

PL UID:

## Question 1: heap\_shorts\_array\_implementation\_mc

---

{ "heap\_array": [ { "key": "a", "html": "\n    The third-smallest element is at either  $\text{arr}[2]$  or  $\text{arr}[3]$ . \n    }, { "key": "b", "html": "\n    Using the heap to sort the array is asymptotically as efficient as using mergeSort. \n    }, { "key": "c", "html": "\n    Swapping the last and the second last element in the array results in another valid heap. \n    }, { "key": "d", "html": "\n    The heap order property may be violated by swapping the contents of two adjacent locations in  $\text{arr}$  so that \n    the first is now smaller than the second. \n    } ] }

---

### Question 1.1: heap\_array

---

[ { 'key': 'b', 'html': '\n    Using the heap to sort the array is asymptotically as efficient as using mergeSort. \n    }, { 'key': 'd', 'html': '\n    The heap order property may be violated by swapping the contents of two adjacent locations in  $\text{arr}$  so that \n    the first is now smaller than the second. \n    } ]

---

## Question 2: heap\_shorts\_build\_heap

---

```
{"input": {"_type": "dataframe", "_value": {"data": [[85, 32, 88, 61, 52, 5, 46, 37, 13, 72]], "index":  
["key"], "columns": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}}
```

---

### Question 2.1: arr

---



---

```
[5, 13, 46, 32, 52, 88, 85, 37, 61, 72]
```

---

### Question 3: heap\_shorts\_operations

---

```
{"graph": "strict digraph {\n\tnode [label=\"\\N\\"]; \n\t0\t [label=16]; \n\t1\t [label=21]; \n\t0 -> 1; \n\t2\t [label=20]; \n\t0 -> 2; \n\t3\t [label=54]; \n\t1 -> 3; \n\t4\t [label=53]; \n\t1 -> 4; \n\t5\t [label=23]; \n\t2 -> 5; \n\t6\t [label=92]; \n\t2 -> 6; \n\t7\t [label=89]; \n\t3 -> 7; \n\t8\t [label=95]; \n\t3 -> 8; \n\t9\t [label=83]; \n\t4 -> 9; \n\t10\t [label=59]; \n\t4 -> 10; \n} \n", "removed_values":  
16  
20  
21  
23  
53  
54  
59  
83  
89  
92  
95", "inserted_values":  
16  
20  
21  
23  
53  
54  
59  
83  
89  
92  
95", "heap_manipulation_insert": [{"key":  
"a", "html": "16"}, {"key": "b", "html": "20"}, {"key": "c", "html": "21"}, {"key": "d", "html": "23"},  
{"key": "e", "html": "53"}, {"key": "f", "html": "54"}, {"key": "g", "html": "59"}, {"key": "h", "html":  
"83"}, {"key": "i", "html": "89"}, {"key": "j", "html": "92"}, {"key": "k", "html": "95"}],  
"heap_manipulation_remove": [{"key": "a", "html": "16"}, {"key": "b", "html": "20"}, {"key": "c",  
"html": "21"}, {"key": "d", "html": "23"}, {"key": "e", "html": "53"}, {"key": "f", "html": "54"}, {"key":  
"g", "html": "59"}, {"key": "h", "html": "83"}, {"key": "i", "html": "89"}, {"key": "j", "html": "92"},
```

```
{"key": "k", "html": "95"}}
```

---

Question 3.1: heap\_manipulation\_insert

---



Expected: [{ 'key': 'c', 'html': '21'}, { 'key': 'e', 'html': '53'}, { 'key': 'g', 'html': '59'}]

---

### Question 3.2: heap\_manipulation\_remove

---

---

Expected: [{key: 'a', 'html': '16'}, {key: 'b', 'html': '20'}, {key: 'd', 'html': '23'}, {key: 'g', 'html': '59'}]

---

## Question 1: disjt\_unions

---

```
{"n": 15, "arr": {"_type": "dataframe", "_value": {"data": [[3, -1, 14, -11, 14, 14, 7, -3, 3, 7, 3, 12, 3, 12, 3]], "index": ["arr"], "columns": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]]}, "bigdata": 82309, "bigunions": 44123}
```

---

### Question 1.1: height

---

---

Expected: 2

---

Question 1.2: unions

---

---

---

Expected: 12

---



Question 1.3: uptreenum

---

---

Expected: 38186

---

#### Question 1.4: files

---

---

playListAns.cpp

---

This is a sample student answer.

---

This is a blank page.

mdtest.md

---

This is a sample student answer.

---

This is a blank page.

picture.png

---

This is a sample student answer.

---

This is a blank page.