

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 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 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  node [label=\"\\N\\"]; \n  0 [label=16]; \n  1 [label=21]; \n  0 -> 1; \n  2 [label=20]; \n  0 -> 2; \n  3 [label=54]; \n  1 -> 3; \n  4 [label=53]; \n  1 -> 4; \n  5 [label=23]; \n  2 -> 5; \n  6 [label=92]; \n  2 -> 6; \n  7 [label=89]; \n  3 -> 7; \n  8 [label=95]; \n  3 -> 8; \n  9 [label=83]; \n  4 -> 9; \n  10 [label=59]; \n  4 -> 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.

picture.png

This is a sample student answer.

This is a blank page.

This is a sample student answer.

This is a blank page.