Set in c++

Assignment Solutions

PW SKILLS

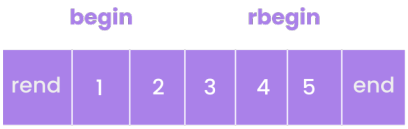
Q1. Consider the following statements:

a. begin() returns an iterator that points at the smallest element of the set.
b. rbegin() returns an iterator that points at the greatest element of the set.
c. end() returns an iterator that points to the next location after the greatest element of the set.
d. rbegin() returns an iterator that points to the next location after the greatest element of the set.
e. rend() returns an iterator that points to the location before the smallest element of the set.
f. rend() returns an iterator that points to the location after the smallest element of the set.

Now choose the correct option:

a. A,B,C,E
b. A,B,C,F
c. A,C,D,E
d. A,B,C,D,F
e. A,B,C,D,E

Answer: a
Explanation: Look at the following diagram:
Consider the set s = {1, 2, 3, 4, 5}



From the diagram we can clearly see the location at which the iterators returned by the following functions point to. So, now you can conclude that statements A,B,C,E are correct. Notice, since set is always in sorted order so first element = smallest element and last element = greatest element.

Q2. Consider the following statements to initialize a set:

```
a. set<int>s1({1,2,3,4,5,6,7,8});
b. int a[] = {6,7,8,9};
   set<int>s2(a.begin(), a.end());
c. int a[4] = {6,7,8,9};
  set<int>s3(a.begin(), a.end());
d. int a[5] = {6,7,8,9};
   set<int>s4(a, a+4);
e.vector<int>v = {4,5,6};
  set<int>s5(v.begin(), v.end());
 f. set<int>s6(s5);
g.set<int>s8(begin(s1), end(s1));
h.set<int,greater<int>>s8;
i. set<int,greater<int>>s9(s1);
```

Which of the following methods of initialization of set in C++ are incorrect?

a. All are correct
b. B,C,I
c. B,C,D,I
d. A,B,D,E,G,H
e. D,E,F,G,H,I
f. A,C,E,H,I
g. A,F,G,H,I

Answer: b
Explanation: Here, the statement B,C,I for initializing set will throw an error.
B) int a[] = {6,7,8,9};
set<int>s2(a.begin(), a.end());
Here, a is an array, so we can't use begin() or end() over here. However, we can use begin() and end() for vectors or sets as used in E and G which is correct.

C) int a[4] = {6,7,8,9};
set<int>s2(a.begin(), a.end());
Here, a is an array, so we can't use begin() or end() over here. The only difference between option B and C is a[] and a[4] which makes no difference.

I) set<int,greater<int>>s9(s1);
Here, we are creating a set s9 in which elements will be sorted in strictly decreasing order, however set s1 is created by default in ascending order. Thus, the an error will be thrown if we try to do this.

Rest all the options are correct way of initializing a set. A,D,E,F,G,H are correct.

Q3. What will be the output of the following code:

```
int main(){
  set<string>s;
  string str = "coding is the best";

  for(int i = 0; i < str.size(); i++){
    string temp;
    while(str[i] != ' ' && i != str.size()){
      temp += str[i];
      i++;
    }
    s.insert(temp);
  }
  for(auto it : s)
    cout<<it<<"*";
}
```

a. coding*is*the*best
b. coding*is*the*best*
c. best*coding*is*the*
d. bes*coding*is*the*

Answer: c

Explanation: This code is taking each word of string str, and then it is storing each word in the set s of type string. So, set 's' will be like:
{"best" , "coding" , "is" , "the"}
Note that in a set elements are always in sorted order. So, by default even if the set is of type string, elements will still be present in sorted order i.e. according to alphabetical order. And while printing these, first the string is printed and then *. So the correct answer will be
=> best*coding*is*the*

Q4. Given two vectors. You have to find the union of these two vectors.
The union contains all the unique elements of two or more vectors.
You can return the answer in any order.
Example-1
Input:
n1 = 6
v1 = [1,1,2,3,2,4]
n2 = 3
v2 = [4,5,6]
Output:
[1,2,3,4,5,6]

Example-2
Input:
n1 = 4
v1 = [1,1,0,0]
n2 = 5
v2 = [9,9,9,9,8]
Output:
[0,1,8,9]

Answer: Code: https://pastebin.com/8NZLcfkN

Code explanation: Here we are traversing through v1 and inserting its elements to set s. Same we are doing for v2. Now, the set s is having elements of both v1 and v2, and since the set only stores unique elements so there won't be any repetition. Thus, set s will be our union of v1 and v2.

Output:

```
n1 = 6
v1 = 1 1 2 3 2 4
n2 = 3
v2 = 4 5 6

The union of v1 and v2 is: 1 2 3 4 5 6
```

```
n1 = 4
v1 = 1 1 0 0
n2 = 5
v2 = 9 9 9 9 8

The union of v1 and v2 is: 0 1 8 9
```

Time complexity: O(n1+n2) ~ O(n) Linear time complexity
Here we are traversing each vector at least once and insert their elements into the set.

Space complexity: O(n1+n2) ~ O(n) Linear space complexity
Since we are creating a set and storing unique elements of both vectors v1 and v2.

Q5. Given two vectors. You have to find the intersection of these two vectors.
The intersection contains all the unique elements that are common between the two or more vectors.
It is possible that the vectors have no element in common.
You can return the answer in any order.
Example-1
Input:
n1 = 6
v1 = [1,1,2,3,2,4]
n2 = 3
v2 = [4,5,6]
Output:
[4]

Example-2
Input:
n1 = 5
v1 = [1,2,3,4,1]
n2 = 7
v2 = [1,1,3,9,2,5,4]
Output:
[1,2,3,4]

Q1. What will be the output of the following code:

```
int main(){
  set<int>s = {615, 67, 4, 19, 10, 89};
  auto it = s.begin();
  cout<<*it<<" ";
  it = s.end();
  it--;
  cout<<*it;
}
```

a. 615 89
b. 615 10
c. 4 615
d. 4 89

Q2. Consider the following code statements for removing element(s) from a set.
s = {1,2,3,4,5,6,7,8}

a. s.erase(8);
b. s.erase(*s.begin()+5);
c.

```
auto start = s.begin();
start++;
auto end = s.end();
end--;
s.erase(start, end);
```

d. s.clear();

Which of the above statement(s) will remove element(s) from the set?
a. A,C,D
b. A,C
c. A,B,D
d. A,B
e. B,D
f. B,C,D
g. All the statements

Q3. Consider the following code:

```
int main(){
  set<int> s = {22, 34, 78, 99, 3, 23, 15, 1, 89};

  auto it1 = s.begin();
  auto it2 = s.end();
  it2--;

  auto it3 = s.rbegin();
  auto it4 = s.rend();
  it4--;
}
```

If the above code is working perfectly fine, choose the correct option:
a. *it1 == *it4 and *it2 == *it3
b. *it1 == *it3 and *it2 == *it4
c. *it1 == *it2 and *it3 == *it4
d. None of the options

Q4. Given an array of size n filled with natural numbers in random order. The array has only one repeating element. The task is to find that repeating element.

Input1:
n = 7
a = [1, 3, 2, 3, 4, 8, 9]
Output1:
3
Input2:
n = 9
a = [4, 5, 9, 2, 3, 4, 89, 11, 15]
Output2:
4

Q5. There is a person who is at current_pos position and a binary string path which is the moves the person took, if path[i] = '0' then the person moved one step left, and if path[i] = '1' then the person moved one step to the right. The task is to find the count of distinct positions the person visited.

Input1:
current_pos = 5
path = "011101"

Output1:
4
Explanation:
Given moves are left, right, right, right, left, and right
i.e. 5 -> 4 -> 5 -> 6 -> 7 -> 6 -> 7
The number of distinct positions is 4 (4, 5, 6, and 7).

Input2:
current_pos = 3
path = "110100"
Output2:
3
Explanation:
Given moves are right, right, left, right, left, and left
i.e. 3 -> 4 -> 5 -> 4 -> 5 -> 4 -> 3
The number of distinct positions is 3 (3, 4, and 5).

Answer: Code: https://pastebin.com/JdPqkgHN

Code explanation: First we are traversing vector v1 and storing its elements in set s. Next we are traversing on vector v2 and checking if an element of v2 is present in set s or not. If it is present then, then that element is common between v1 and v2, so we add that element to ans set. And at the end, we print the ans set.

Output:

```
n1 = 6
v1 = 1 1 2 3 2 4
n2 = 3
v2 = 4 5 6

The intersection of v1 and v2 is: 4
```