std::deque

09 June 2021 07:07

1. ## std::deque double ended queue - Creation

```
EMPTY    - std::deque<int> first;                                 // empty deque of ints
#ELEMENTS - std::deque<int> second (4,100);                        // four ints with value 100
PARTIALCP- std::deque<int> third (second.begin(),second.end());  // iterating through second
FULL COPY- std::deque<int> fourth (third);                        // a copy of third
 // the iterator constructor can be used to copy arrays:
int myints[] = {16,2,77,29};
```

2. ## Iterators – const, non-const random-access and reverse iterator

3. ## Members – front, back, begin, end, rbegin, rend, c(r)begin, c(r)end, size, empty [], at

4. ## Size

   - Size - number of elements currently used
   - Empty?

   MAX_SIZE - ???

5. ## Capacity changing

   - shrink_to_fit () - make capacity = size

6. ## Deque size related

   - resize(#, value) - if #>size insert at end with value value or val init.. if #<size destroy elements at end? making size = n
   - erase(it) or erase(itr1, itr2) - and reduce size
   - clear() - make size 0
   - pop_front, pop_back

   - assign(#, value) or assign(itr1, itr2) where itr1, itr2 is from different container
   - insert(itr, #, value) or insert(itr, itr1, itr2), push_front, push_back
   - emplace(itr, constructor args), emplace_front, emplace_back,

7. ## Operator =, []

   I

8. ## F

   I

9. ## F

   I

10. ## F

11. ## F

    I

12. ## F

    I

13. ## F

    I

14. ## F

    I

15. ## F

    I

16. ## F

    I

17. ## D

    I

18. F
    I
19. F
    I
20. F
    I
21. F
22. F
    I
23. F
    I
24. F
    I
25. F
    I
26. F
    I
27. F
    I
28. D
    I
29. F
    I
30. F
    I
31. F
    I
32. F
33. F
    I
34. F
    I
35. F
    I
36. F
    I
37. F
    I
38. F
    I
39. D
    I
40. F
    I
41. F
    I
42. F
    I
43. F
44. F
    I
45. F

I

46. F

    I

47. F

    I

48. F

    I