Week 3: Lab

Collaboration level 0 (no restrictions). Open notes.

1. Consider the linear-time merge algorithm discussed in the notes and a possible implementation below. How many element comparisons will the standard merge function take to merge the following left and right lists ?

   $left = [1, 3, 4, 5, 6, 7, 8]$, $right = [1, 5, 9, 11, 12, 16]$

   Merge(left, right)

```
result = []
i=0
j=0
while i < len(left) and j < len(right):
     if left[i] < right[j]:
          result.append(left[i])
          i = i+1
   else:
        result.append(right[j])
        j = j+1

  # add any left overs
  while i < len(left):
      result.append(left[i])
      i = i+1
  while j < len(right):
      result.append(right[j])
      j = j+1

  return result
```

   A 8

   B 9

   C 10

   D 13

Find a $\Theta()$ bound for the following recurrences using iteration. Assume $T(1) = 1$.

What we expect: show $O(1)$ steps of your iteration, with the general formula, the recursion depth, and the final $\Theta()$ bound for $T(n)$. **Do not write your answers on this page.** *Use a separate page for each problem and show your work.*

2. $T(n) = T(n/2) + 1$
3. $T(n) = T(n/3) + 1$ (assume $T(i) = 1$ for $i = 1, 2$).
4. $T(n) = T(n/10) + 1$ (assume $T(i) = 1$ for $i < 10$).
5. $T(n) = T(2n/3) + 1$ (assume $T(i) = 1$ for $i = 1, 2$).
6. $T(n) = T(n - 1) + 1$
7. $T(n) = T(n - 2) + 1$ (assume $T(i) = 1$ for $i = 1, 2$).
8. $T(n) = T(n - 3) + 1$ (assume $T(i) = 1$ for $i = 1, 2, 3$).
9. $T(n) = T(n/2) + n$
10. $T(n) = T(n/3) + n$ (assume $T(i) = 1$ for $i < 3$).
11. $T(n) = 3T(n/3) + \Theta(n)$ (assume $T(i) = 1$ for $i < 3$).
12. $T(n) = 5T(n/5) + \Theta(n)$ (assume $T(i) = 1$ for $i < 5$).
13. $T(n) = T(n - 1) + n$
14. $T(n) = T(n - 2) + \Theta(n)$ (assume $T(i) = 1$ for $i = 1, 2$).
15. $T(n) = 2T(n - 1) + \Theta(1)$

16. $T(n) = T(\sqrt{n}) + 1$

17. $T(n) = 7T(n/2) + n^3$

18. $T(n) = 7T(n/2) + n^2$

19. $T(n) = T(n-1) + 2n - 3$, with $(T(1) = 1$

20. $T(n) = 2T(n-1) + 1$, with $(T(1) = 1$

21. $T(n) = 6T(n/6) + 2n + 3$, with $(T(1) = T(2) = T(3) = T(4) = T(5) = 1$

22. $T(n) = 4T(n/3) + 2n - 1$, with $(T(1) = T(2) = 1$

23. $T(n) = 3T(n/2) + n^2$, with $(T(1) = 1$

24. $T(n) = 7T(n/2) + n^2$, with $(T(1) = 1$

25. (challenge) $T(n) = T(n/3) + T(2n/3) + \Theta(n)$ (only guess the solution)

26. (challenge) $T(n) = T(n/3) + T(n/4) + \Theta(n)$ (only guess the solution)

27. (challenge) $T(n) = T(n/2) + T(n/4) + T(n/10) + \Theta(n)$ (only guess the solution)

Partial Answers

1. B (9 comparisons)
2. $\Theta(\lg n)$
3. $\Theta(\lg n)$
4. $\Theta(\lg n)$
5. $\Theta(\lg n)$
6. $\Theta(n)$
7. $\Theta(n)$
8. $\Theta(n)$
9. $\Theta(n)$
10. $\Theta(n)$
11. $\Theta(n \lg n)$
12. $\Theta(n \lg n)$
13. $\Theta(n^2)$
14. $\Theta(n^2)$
15. $\Theta(2^n)$
16.
17.
18.
19. $T(n) = T(n-1) + 2n - 3$, with $(T(1) = 1 : T(n) = \Theta(n^2)$
20. $T(n) = 2T(n-1) + 1$, with $(T(1) = 1$: $T(n) = \Theta(2^n)$ Note: For exponential recurrences we are usually happy with just a lower bound.
21. $T(n) = 6T(n/6) + 2n + 3$, with $(T(1) = T(2) = T(3) = T(4) = T(5) = 1$: $T(n) = \Theta(n \lg n)$
22. $T(n) = 4T(n/3) + 2n - 1$, with $(T(1) = T(2) = 1$: $T(n) = \Theta(n^{log_3 4})$
23. $T(n) = 3T(n/2) + n^2$, with $(T(1) = 1$: $T(n) = \Theta(n^2)$
24. $T(n) = 7T(n/2) + n^2$, with $(T(1) = 1 : T(n) = \Theta(n^{\log_7 8})$