

Name: Reza Shisheie

Login ID: 2708062

HW1 Reza Shisheie

1. We can sum $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ inside text or use the displayed math like below:

$$\sum_{k=1}^{n} k = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

2. The recurrence (1) shows the worst-case running time T(n) of mergesort:

$$T(n) = \begin{cases} c & \text{if } n = 1, \\ 2T(n/2) + cn & \text{if } n > 1. \end{cases}$$
 (1)

Using the master theorem in Chapter 4, we can get $T(n) = \Theta(n \log n)$.

3. The recurrence (2) shows the worst-case running time T(n) of binary search:

$$T(n) = \begin{cases} c & \text{if } n = 1, \\ T(n/2) + c & \text{if } n > 1. \end{cases}$$
 (2)

Using the master theorem in Chapter 4, we can get $T(n) = \Theta(\log n)$.

4. Browse https://www.cs.dartmouth.edu/~thc/clrscode/clrscode3e.pdf to learn how to use the clrscode3e package in LaTex to typeset pseudocode.

INSERTION-SORT(A)

1 for j=2 to A. length2 key=A[j]3 # Insert A[j] into the sorted sequence $A[1\mathinner{.\,.} j-1]$.

4 i=j-15 while i>0 and A[i]>key6 A[i+1]=A[i]7 i=i-18 A[i+1]=key

If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$