**Homework 1**

**Problem 1: (Asymptotic Growth)** All the unstated logs are in base 2.

**n2n, nlog(n), n!, n, n100, 2n, log(n), log(n!), (log(n))!, 22^n.**

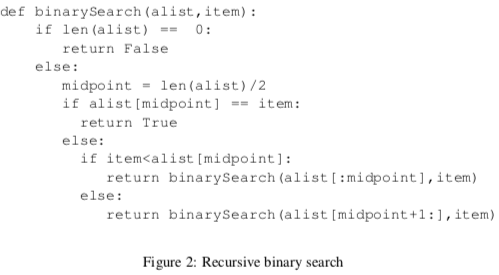
* g1=log(n)
* g2=n
* g3= log (n!)
* g4=nlog(n)
* g5=n100
* g6=(log(n))!
* g7=2n
* g8= n2n
* g9= n!
* g10=22^n

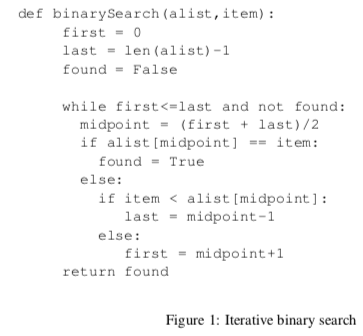
g1 = O(g2), g2 = O(g3), ..., g9 = O(g10)

**Problem 2: (Recurrences)** All the unstated logs are in base 2.

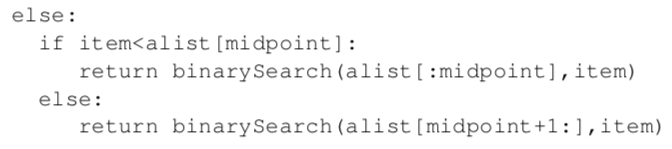
* **(a)** T(n) = 2T(n/2) + n3 solution by master theorem,
  + a=2, b=2;
    - n^logba= n, f(n)= n3
    - If ; f(n) = Ω(nε) then ε= 3 and 2(n/2)3 ≤ cn3 for c = ¼;
      * **T (n) = Θ(n3) (case 3)**
* **(b)** T(n) = 7T(n/2) + n2 solution by master theorem,
  + a=7, b=2;
    - n^logba= nlog7, f(n)= n2
    - If ; f(n) = O(n2+log3-ε) then ε=log3;
      * **T (n) = Θ(nlog7) (case 1)**
* **(c)** T(n) = 2T(n/4) + n solution by master theorem,
  + a=2, b=4;
    - n^logba= n^log42, f(n)= n1/2
    - If ; f(n) = Θ(n1/2logk+1n) then k=0;
      * **T (n) = Θ(n1/2 log n) (case 2)**
* **(d)** T(n)=T(n−1) + n solution by substitution,
  + With an educated guess, T(n-1) is iterates n times and in each iteration it’s computational complexity is n;
    - Guess = O(n2), assume that T(k) ≤ ck2 for k<n;
    - Prove; T(n) ≤ cn2 solution by induction;
      * T(n)=T(n-1) + n
      * ≤ c(n-1)2 +n = c(n2–2n+1) +n = cn2-(c(2n-1)-n);
        + ≤ cn2 when, (2n-1)c-n ≥0 or (2c-1)n-c ≥0; c≥1 and n≥1

**T (n) = O(n2)**

**Problem 3: (Binary- Search Python)** All the unstated logs are in base 2.



* **(a)**
  + **(i)**
    - By “midpoint = (first + last)/2” statement, function reduces length of the list to half in each iteration, because of that situation upper bound of the function is **O(logn).**
    - In the worst case (item doesn’t exist inside the list), asymptotic running time of the algorithm is **O(logn).**
  + **(ii)**
    - Recurrence is **T(n/2) + n** because,



If and else part cause “T(n/2)” and copy opetartion by “alist[:midpoint]” cause to “n”. Solution by master theorem,

* + - * a=1, b=2;
      * n^logba= n^log12, f(n)= n dominates.
        + **T (n) = O(n) (case 3)**