**Homework (Search)**

**Linear(chiziqli search)** ----> Chiziqli search bo`lib , u malumotni qidirishda oddiy yaniy ketma-ket qidirish algoritmidan foydalanadi. Shuning uchun ham qidiruvdan foydalanganimizda eng yomon holatlardagina bu searchdan foydalanamiz. Shu bn birga bu search qidirish uchun sarflaydigan vaqt O(n) ga teng.

Bu yerda bu yerda n qadamlar soni

Masalan: 1,2,3,4,5,6,7,8.9 sondan (7) ni topish uchun bu search 7 ta qadam sarflaydi.

**Binary search** ----> Bu search da misol bilan tushuntirsak masalan :

Bizda 10,20,30,40,50,60,70 kabi malumotlarimiz bor. Bizga bu malumotlardan 60 soni kerak bulsin , shunda bu search bu malumortlar ichidan urtachasini tanlab oladi. Yaniy bu holatda 40 ni tanlab olib (biz qidirayotgan malumot 40 dan kata yoki kichikligiga tekshiradi va ) bizga kerak bulgan sondan chap tarafdagilarni (10,20,30) tashlab yuboradi va o`ng tomondagilar bilan biz so`rogan sonni (=,>,<) so`rovlarga tekshiradi. Hamda bu search faqatgina tartiblangan malumotlar bilan ishlashdagina qo`l keladi. Boshqa holatlarda esa bu search eng yomon search hisoblanadi. Hamda bu search log2(n) vaqt sarflaydi . bu yerda n qadamlar soni

**Array search [List search]** ----> Array search agar biz malumot indeksini bilsak va uni o`qimoqchi bulsak, foydali. Lekin agar biz Listdagi biron-bir malumotni uchirmoqchi,update yoki qushmoqchi bulsak not useful hisoblanadi. Sababi agar biz pythonga murojaat qilsak. Agar biz biron-bir elementga uchirsak yoki qushsak list elementlarining xotiradagi index si uzgaradi. Yaniy qushsak o`ng tarafga , ayirsak chap tarafga qarab surulardi. Shuning uchun ham read uchun O(1) vaqt sarflaydi. List kurinishida variable yaratsak, shu variable hajmicha xotiradan malum bir joy topiladi.Agar joy topilmasa Memory Error degan xato qaytaradi. Va add hamda remove qilish uchun biz O(n) vaqt sarflaymiz.

**Linked List --->** Yaniy bog`langan list ham deb ataladi. Bu searchda bizda tugunlar , bog`langan tugunlar buladi. Bu tugunlarda malumotlar ushbu korinishda

saqlanadi. Shu bilan birga malumotlar tartiblanmagan lekin bog`langan holda xotiraning istalga joyida saqlanadi.

Hamda bu searchda malumotni qushishda, yoki uchirishda qanaqadir index uzgartirishlarga vaqt sarflaymiz. Demak bu search malumot qushish va ayirishda useful roq ekan . Ammo biz agar malumotni read qilmoqchi bulsak usha malumotdan oldingi va keyingi malumot id sini bilishimiz kerak. Shuning uchun ham bu search not readeable ekan.