Data strukuturlar ve algoritm

1)ESAS ANLAYISLAR

Algoritm nedir-Alqoritm 9-cu əsrdə Cəbrin əcdadı və qurucusu kimi tanınan Harezminin cəbr sahəsində apardığı araşdırmalar nəticəsində ortaya çıxıb. “Alqoritm nədir?” sualının cavabı məsələnin həlli üçün tələb olunan addimlarin butunune verilən addır. Xülasə, alqoritmlər müəyyən bir vəziyyətdən başlayaraq müəyyən nəticə ilə bitən problemlərin həllini təmin edən addımlardır.

2) İnformasiyanın kompüterdə ifadəsi

KOmputerde informasiyalar 0 ve 1 seklinde saxlanirlir. 0 ve 1 den ibaret olan system binary system adlanir(ikili system). İkili ədədlərdəki 1 və 0 (bit) rəqəmləri kompüterin elektrik enerjisini ötürmək üçün istifadə etdiyi tranzistorların açıq və ya sönük olduğunu göstərir. Transistorlar iki təlimata malikdir, 0 (söndürmək) və 1 (açmaq).

3)Say sistemleri

Ədədləri ifadə edərkən adətən onluq say sistemindən istifadə edirik. Onluq say sistemində diapazonumuz 0-dan 9-a qədər olduğundan, biz 10 müxtəlif vəziyyəti bir rəqəmlə ifadə edə bilərik. Binar sistemdə iki fərqli vəziyyət bir rəqəmlə ifadə edilir; 0 və 1.!!!!! Bir rəqəmlə!!!!!

4)Melumatlarin saxlanmasi

**Bayt** (ing. byte) — əsas informasiya vahidi olub 8 bitə bərabər hesab olunur. Rəqəmli informasiyanın saxlanması və emal olunması vahididir.

4)Recursion-bpyuk prob;emleri kicik porblemlere bolerek hell etmemimzi saglayan bir yapi.

Fibonacci ededi-0; 1; 1; 2; 3; 5; 8; 13; 21; 34; 55; 89...

Tərifə əsasən, ilk iki Fibonaççi ədədləri 0 və 1-dir. Sonra gələn ədəd özündən əvvəlki ilk iki ədədin cəminə bərabərdir. Bəzi mənbələrdə sıranın ilk ədədi 0 yox, 1 götürülür.

Riyazi dildə, Fibonaççi sırası *Fn* aşağıdakı rekurrent düsturla verilir

2. MELUMAT STRUKTURLARİ

1)Array-Massiv

Massivlərə məna vermək üçün birdən çox obyekt lazım ola bilər. Məsələn, indi qarşınızda olduğunuz kompüter nümunəsini nəzərdən keçirək. Masaüstü kompüterlər klaviatura-siçan-monitor triosunu birləşdirdikdə məna kəsb edir. Onların heç biri olmadan əməliyyat etmək olar, lakin çətindir.

Array-ın çatışmazlıqlarından biri olan yaddaş problemini araşdıraq. Gəlin kompüter nümunəmizlə davam edək. Bizdə artıq klaviatura, siçan və monitor var. Yeni bir monitor alarkən bizə daha böyük bir masa lazımdır. Eyni şey yeni klaviatura və ya siçan aldığımız zaman da gedir. Bir yerdən başqa yerə köçərkən vaxt və enerji itiririk.

Dinamik massivlər isə yeni element üçün boş yer saxladıqları üçün çevikdir. Məsələn, bəzi mətbəx masaları açılan sürüşmə quruluşuna malikdir. Masanı kiçik olduqda böyütmək üçün istifadə olunur. Dinamik massivlər eyni məntiqə malikdir. Onlar yeni işçilər üçün yer tuturlar.

Dynamic Array-ın çatışmazlıqlarından biri yaddaşda əlavə yer tutmasıdır ki, bu da başqa hadisənin baş verməsinin qarşısını ala bilər. Bir nümunə ilə necə başa düşək. Cədvəl nümunəsi haqqında danışdıq. Biz əlavə yer tutduq və qonaqlarınızın problemi ola biləcəyi üçün mobilliyimizi itirdik.

Həmişə mənfi cəhətlərdən danışırsınız, amma bunun bir üstünlüyü yoxdur? Əlbəttə var. Massivlərin qarşılıqlı əlaqəsi onlara daxil olmağı asanlaşdırır. Keyboard-Mouse-Monitor nümunəsini verdik. Hamısı bir masada olanda ona çatmaq asandır. (Cədvəl = Massiv, Klaviatura-Maus-Monitor = Massiv Elementi).

Array-da meumatlsr tutlan zaman yanyana olmalidir ardarda.

2)Linked List- bizə məlumatları yan-yana saxlamadan yadda saxlamağa imkan verən strukturlardır. Yeni element üçün yaddaşda yeni boşluq açmağa ehtiyac yoxdur. Array-dan fərqli olaraq, elementlər yaddaşda dagila ve ya paylana(qarisiq sekilde) bilər, lakin sonuncu element öz ünvanının əvvəlki elemente xəbərdar etməlidir. Hər bir əvvəlki element növbəti elementə bağlanır.

3)Linked Lİst vs Array

Linked Lİst-elementler ozunden sonra gelen elemnetlerin lokasyoununu da tutur. Qarisiq shekilde tututlur.Linked Listlerde istenilen elemente ulasmaq ucun hrmin elemnetin ozunden evvelki elemente catmaq lazimdir. Linkedin listde elementden elave, ozunden sonraki elementi ve onun lokasionunu tutur.Linkedin lsitlerde yeni element daxil etmek ve ya ortadan element silmek arrayalar gore daha rahatdir.

Array-melumatlar yanyaana tutulut. Arrayin istenilen elementine ulasmaq sabit vaxtda oıur. İstenilen elemente aninda ulasmaq mumkundurş buna random access deyilir. Sadece element tuutldugu ucun burada dahaa az yer tutur.

��=��−1+��−2,

​