C# Data structure & Algorithms Summery part-1 By Mahmoud Badawy

-----------------------------------------------------

ملخص Data structure

MB

##################################################################

C# Data structure & Algorithms

-----------------------------------------------------

ايه هى Data structure ؟

الداتا ستراكتشار طريقه ترتيب و تخزين البيانات بتعتى و دى بتساعدنى انى احسن من برفورمانس الكود و بتنقسم لنوعين

Linear & non Linear و دى صوره بتوضح بيتكونوا من ايه بالظبط

Linear DS : بتنقسم اللينيار داتا ستراكتشار الى static Ds و دى بتحتوى على ال Array

و dynamic DS و دى بتحتوى على Queue & Stack & linked list

Non Linear DS : Tree & Graph

A diagram of data structure

Description automatically generated

ايه هى Algorithms ؟

طرق حل المسائل البرمجيه لان المساله البرمجيه بتتحل بعده الطرق و لكن الخوارزميات بتخليك تكتب الكود بشكل علمى يراعى الوقت الى هيتنفذ فيه الكود و جودته performance

ايه هى Algorithm Analysis ؟

دى طريقه تحليل الالجوريزم بحيث عن طريقها نعرف الكود او الالوجوريزم الى هنفذه ده هياخد وقت اد ايه و البرفورمانس بتاعه كويس و لا لأ قبل بدايه كتابه الكود و على اساس الحسابات دى هشوف انا عاوز اوفر فى الوقت ولا المساحه و اشتغل على اساسه Space Or time من الاخر لان ممكن يكون يهمنى سرعه البرنامج ميهمنيش المساحه لان السيرفر كبير جدا مثلا او عندى ميمورى كبيره جدا فمش فارق معايا و كذلك فى وقت تانى بيبقى عندى الميمورى صغيره خالص و لازم احافظ على المساحه فساعتها بحاول اعمل حاجه تكون بتستخدم اقل قدر ممكن من مساحه الميمورى و هكذا بقه

ايه هى ال Cases بتاعت الاناليسيس لاى الجويرزم ؟

Best Case & Average Case & Worst Case

طيب اذاى بيتم قياس جوده الالجوريزم ؟؟

بيتم قياسها بالوقت الى الكود هياخده قياسا على حجم ال input الى داخل للالجوريزم و ده بيتم بفانكشنز ومعادلات رياضيه خاصه بالايفالويشن ده

طيب ايه الفانكشنز بيتم قياس جوده الالجوريزم بيها ؟؟

ترتيبهم حسب سرعه البرامج الى بتتعمل بيهم و للحسابات اعتبر ان ال n = 8 مثلا و ال n دى هى عدد اللفات الى الكود هيلفها على فانكشن معينه

**Constant** : F(n) = 1 الاسرع و الاكثر توفيرا على الاطلاق هنا الكود مش هيلف فى حته كمثال int x = 1;

int[] array = {1, 2, 3, 4, 5};

int element = array[2]; // O(1)

logarithmic : F(n) = log(n) و دى عباره عن عمليات قسمه متكرره و لذلك هتلاخظ بعد كده انها اقل من ال n لوحدها فى استهلاك الميمورى و اكبر فى السرعه و ده لان مثلا لو انا عندى ال n = 8 و انا بقوله انى عاوز اعمل عدد اللفات دى بس بدل ما يعملها كلها لأ انا عاوزه يمشيلى اللفات دى بدل خطوه خطوه يبقى خطوات اسيه فيبقى عدد اللفات الفعليه الى هيعملها 3 لفات بدل 8 فكده هيتعمل فى 3 ثوانى مش 8 ثوانى مثال هنا بقوله يا اسطى فى كل لفه على اللوب اضرب ال n فى 2 المره الاولى 1 المره التانيه 2 المره التالثه 4 المره الرابعه هلاقى نفسه ب 8 فمش هيكمل اصلا لانه انا قايله اصغر من 8 كده خلص فى 3 ثوانى

for(int n = 1 ; n < 8; n \* 2)

بص من الاخر طالما اللوب فيها \* ضرب او / قسمه عدد الخطوات فخلاص يبقى دى log(n) لو + جمع او- طرح يبقى linear = O(n) و هتعرفها قدام

و ده لانه مش هيمشى كله برضوا بالعكس ده هيمشى نص الخطوات الى المفروض يمشيها

او ان ا اقسم اللول لنصين نص يمين و نص شمال و اخليه يمشى خطوه من اليمين و خطوه من الشمال علشان يوصل للترجيت ذى ال Binary Search

int BinarySearch(int[] array, int target)

{

int left = 0;

int right = array.Length - 1;

while (left <= right)

{

int mid = (left + right) / 2;

if (array[mid] == target)

return mid;

else if (array[mid] < target)

left = mid + 1;

else

right = mid - 1;

}

return -1;

}

linear : F(n) = n و دى بتبقى سرعتها مرتبطه ارتباط كبير بعدد اللفات الى هو n = 8 لفات لو اللفه بتاخد ثانيه فكده ال 8 لفات هتاخد 8 ثوانى مثال

for(int n = 0 ; n =< 8; n++)

or

for(int n = 1 ; n <= 8; n+2)

Example :

int LinearSearch(int[] array, int target)

{

for (int i = 0; i < array.Length; i++)

{

if (array[i] == target)

return i;

}

return -1;

}

بص من الاخر طالما اللوب فيها \* ضرب او / قسمه عدد الخطوات فخلاص يبقى دى log(n) لو + جمع او- طرح يبقى linear = O(n) و هتعرفها قدام

Linear logarithmic : F(n) = n log(n) و دى بتبقى سرعتها أقل من سرعه n لوحدها او log(n) لوحدها لان دلوقتى بالعفل كده لو قولنا ال n = 8 , log(n)=4 اذا ناتج ضربهم هيطلع 32

و دى بيتم تطبيقها فى Merge Sortالمثال بتعها هنا

<https://www.geeksforgeeks.org/merge-sort/>

او مثالى الخاص

for(int n = 0 ; n =< 8; n++)

for(int n = 0 ; n =< 8; n \* 2 )

Quadratic : و دى بتبقى سرعتها بطيئه لانها بتبقى يعتبر عدد اللفات مضروبه فى نفسها فلو عندى 8 لفات يبقى عدد الثوانى الى هيتم فيها التفيذ 64 مثال

و دى بيتم تطبيقها لما يكون معايا For loop جواها For loop لان اول ما تشوف Forloop اعتبرها n و لو جواها for loop فكده هتلف على اللوب الاولى فى اول عنصر هتخش هتلاقى اللوب الى جواها هتلف على كل العناصر الى فيها فبفرض ان انا عندى اللوب الاولى 8 لفات و اللوب التانيه 8 لفات فمع كل لفه فى اللوب الاولى هيلف جوه اللوب التانيه 8 لفات اذا عدد اللفات الكلى بيساوى n\*n = 8 \* 8 = 64 ثانيه

هتلاقى الكلاب ده بيتنفذ فين بقه

هتلاقيه فى Selection Sort & Bubble Sort & Insertion Sort

<https://www.geeksforgeeks.org/insertion-sort-algorithm/>

<https://www.geeksforgeeks.org/selection-sort-algorithm-2/>

<https://www.geeksforgeeks.org/bubble-sort-algorithm/>

<https://www.geeksforgeeks.org/quick-sort-algorithm/>

او مثالى الخاص

for(int n = 0 ; n =< 8; n++)

for(int n = 0 ; n =< 8; n++ )

Cubic : و دى بتبقى سرعتها بطيئه جدا لانها بتبقى يعتبر عدد اللفات مضروبه فى نفسها ثم مضروبه فى نفسها كمان مره فلو عندى 8 لفات يبقى عدد الثوانى الى هيتم فيها التفيذ 512 مثال

و دى بيتم تطبيقها لما يكون معايا For loop جواها For loop جواها For loop لان اول ما تشوف Forloop اعتبرها n و لو جواها for loop فكده هتلف على اللوب الاولى فى اول عنصر هتخش هتلاقى اللوب الى جواها هتلف على العنصر الاول فى اللوب التانيه فهتلاقى جواه for loop تالت قهتخش تلف على كل العناصر الى فيها فبفرض ان انا عندى اللوب الاولى 8 لفات و اللوب التانيه 8 لفات و اللوب التالته 8 فمع كل لفه فى اللوب الاولى هيلف جوه اللوب التانيه 8 لفات و مع كل لفه من التانيه هيلف 8 مرات فى الثالثه اذا عدد اللفات الكلى بيساوى  
 8\*8\*8 = n\*n\*n = 512 ثانيه

هتلاقى الكلام ده بيتنفذ فين بقه

او مثالى الخاص

for(int n = 0 ; n =< 8; n++)

for(int n = 0 ; n =< 8; n++ )

for(int n = 0 ; n =< 8; n++ )

Exponintial : و دى بتبقى سرعتها بطيئه جدا لان ال n هنا تعبر عن عدد اللوبس الموجوده تحت اللوبس بمعنى ان لو قالك فى المعادله دى ان n=4 مثلا فده معناه ان انا عندى لوب ليفل 1 جواها لوب فى ليفل 2 و اللوب التانيه جواها لوب فى ليفل 3 و اللوب الثالثه جواها لوب فى ليفل 4

او مثالى الخاص لو ال n=8 و ال c=3

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

for (int n = 0; n <= 3; n++)

{

}

}

}

}

}

}

}

}

و دى بيتم تطبيقها لما يكون معايا فور لوب جواها فور لوب و كل فور لوب جواها فور لوب و حاجه كده تشوفها تتشل و فى الغالب دى هتشوفها فى حاجه عباره عن لوب جوه لوب جوه لوب خصوصا لما يبقى فى فانكشن جواها ريكيرشن و الريكيرشن ده جواه لوب فعينك ما هتشوف الا النور بقه

و ده بقه شرح مفصل اكتر فى الموضوع ده بالامثله لكل الانواع

<https://www.geeksforgeeks.org/complete-guide-on-complexity-analysis/?ref=next_article>

اذاى بقه بنحسب ان الفانكشن الى انا كاتبها من انهى نوع من الى فوق دول ؟؟

بص يا سيدى انت بتقعد تحسب كل سطر جوه الفانكشن بيديك ايه بالظبط و بعد ما بتخلص بتشوف المعادله كلها و بتاخد اكبرهم فى استهلاك الوقت ليصبح هو ده نتيجه المعاده يعنى بنختار اسوء ما فى المعادله ليعبر عنها

فلو عندى الفانكشن دى مثلا

static void Main(string[] args)

{

int[] Arr = { 5, 6, 8, 7, 8, 9, 7, 1 };

Console.WriteLine(Arr[2]); // O(1)

for (int n = 0; n <= Arr.Length; n++) // O(n)

{

Console.WriteLine(Arr[n]); // O(n)

}

for (int n = 0; n <= Arr.Length; n++) // O(n)

{

for (int x = 0; x <= Arr.Length; x++) // O(n)

{

Console.WriteLine(Arr[x]); // O(n)

}

Console.WriteLine(Arr[2]); // O(n)

}

//So all Equation is : O(1) + O(n) + ( O(n) \* O(n) )

//So Simple equation : O(1) + O(n) + O(n^2) = O(n^2) لأن ده اكبرهم الورست كيس يعنى

//So time complexity for this code is O(n^2) so it is Slow .

}

الفكره ان الكود الى فى سطر لوحده و مش جوه لوب فده وقت تنفيذه ثابت فيبقى O(1)

و اى لوب بيبقى O(n) و اى كود ثابت ذى المعادلات بيبقى واخد نفس عدد تنفيذ اللوب فيبقى O(n) و بنعتبره ولا كانى موجود اصلا لان اللوب نفسه عبر عنه

و لكن اول ما تشوف لوب اخر جوه اللوب ساعتها بنحسب ان O(n) بتاعت اللوب الاب مضروبه فى اللوب الابن O(n) و ده لان كده عدد اللفات بقى بيساوى n\*n = n^2  
طيب احنا فهمنا ان الكود اللوب الى جوه اللوب بنضرب   
طيب لو لوب جنب لوب ؟؟ هنجمع يا رايق بينهم و فى الاخر احنا بنشوف اكتر حاجه بتاخد وقت فى المعادله كلها و نعبر عن المعادله بيها

طب لو عندك if + else if بتاخد منها اكبر معادله فيهم علشان تعبر عنهم كلهم

و لان الجراف هيفهمك جدا فدى رسمه جميله بتلخص كل الكلام الى فوق ده

دى مساله فى السريع تانيه واخد بالك ان اللوب لو مفيهاش n وفيها عدد ثابت بنحسبها ب 1 يا رايق

A close-up of a math formula

Description automatically generated

A graph of different colored lines

Description automatically generated

A diagram of lines and letters

Description automatically generated

و بما ان عندنا مجموعه كبيره من ال Data Strucure جاهزه ذذى ال Array , Stack , Queue , etc فطبعا لازم نعرف ال Time Complexty بتعتهم علشان نعرف مين اسرع من التانى و علشان كده اتقسموا للمعرف علشان نستخدمهم صح لان فى فورقات بينهك كلهم فى السيرش و الديليت و الانسيرت و الاكسيس مش كلهم شبه بعض و ده Time Complexty بال Big O Notation

A screenshot of a computer

Description automatically generated

طيب ايه الفرق بين ال Time Complexity & Space Complexity ؟   
ال time Complexity : ده الى بيشوف الانبوت الى جاى و الكود الى هيتكتب هيخلي الفانكشن تتنفذ فى مده اد ايه بالظبط و كل ما المده قلت كل اما كان احسن

اما ال Space Complexity : ده بيقيس الكود استهلك ميمورى اد ايه منا مروحش استخدم مثلا كل شويه ميمورى ب size كبير تتحط فى الميمورى تاكلها و اقول كده الكود تمام اكيد لأ انا المفروض بستخدم ما يناسب الكود بحيث ميبقاش حجمه كبير ياخد من الميمورى و كذلك المفروض معملش فاريبولز ملهاش لازمه لا لازم يبقى ليها لازمه و ابقى محدد حجم معين متخطهوش و لو قدرت بعد ما اخلص افضى الميمورى يبقى حلو جدا

**ده موقع جميل جدا بيوريك اذاى كل الالجوريزميس شغاله بشكل كرتونى كده جميل و انت بتذاكر مهم جدا تبص عليه**

<https://visualgo.net/en>

MB

##################################################################

How to think to solve problems

---------------------------------------

الالجويزم : هى مجوعه من الخطوات لحل المشاكل البرمجيه بسهوله :   
1- فهم المشكله كويس جدا

2- التفكير فى مثال ملموس

3- تحويل المشكله الكبيره الى مجموعه مشاكل صغيره

4- حل كل المشاكل الصغيره

5- التاكد من انه قد تم حل المشكله بنجاح

طيب اذاى افهم المشكله :  
1- اسال لحد ما تتاكد انك خلاص فهمت المشكله

2- لازم تعرف ال inputs الى هتستخدمها و انت بتحل المشكله

3- لازم تعرف ال outputs الى المفروض تطلع بعد حل المشكله

4- شوف لو علاقه مباشره بين ال input , output ده هيساعدك فى مل حل المشكله بسرعه

MB

##################################################################

How to improve code

-------------------------------

هنا هعلمك اذاى تفكر و هخليك تحل مسأله بطريقتين الطريقه الاولى O(n) و هنعملها تحسين علشان تبقى O(1)

بص المسأله انا عاوز اديك رقم معين و ليكن 5 فهو هياخده و هيجمع 1+2+3+4+5 و يدينى فى الاخر مجموعهم   
فالحل الاول الى هيجى ف دماغك انى اعمل لوب اللينس بتعته هتكون هى الرقم و هخليه يبدا بالاندكس 1 و يفضل يجمع انديكس 1 على انديكس 2 و هكذا لحد ما يوصل ل 5 و يجمعهم على بعض و خلاص فيبقى كده بيساوى فى الاخر 15

public class Program

{

public static void Main()

{

int Res = AddUpTo(5);

Console.WriteLine($”Res = {Res} .”);

}

public static int AddUpTo( int number)

{

int total = 0;

for (int index = 1; index <= number; ++index)

{

total = total + index;

}

return total;

}

}

طيب لو عاوز احلها حل ليه علاقه بالارقام و يعتمد على الرياضه ان الرقم n=5 فهقوله

اللوب بيمشى انديكس 0 ثم 1 ثم 2 ثم 3 ثم 4

و المعادله الرياضيه الصحيحه للحوار ده المفروض 5+4+3+2+1 يعنى

R=(n-0) + (n-1)+(n-2)+(n-3)+(n-4) = (5-0)+(5-1)+(5-2)+(5-3)+(5-4) = 5+4+3+2+1 = 15

public class Program

{

public static void Main()

{

int Res = AddUpTo(5);

Console.WriteLine($"Res = {Res} .");

}

public static int AddUpTo( int number)

{

int total = 0;

for (int index = 0; index <= number; index++)

{

total += (number - index) ;

}

return total;

}

}

ألحل التانى اننا نمشى من اليمين للشمال بنفس الطريقه الرياضيه دى

public class Program

{

public static void Main()

{

int Res = AddUpTo(5);

Console.WriteLine($"Res = {Res} .");

}

public static int AddUpTo( int number)

{

int total = 0;

for (int index = number; index >= 0; index--)

{

total += (number - index) ;

}

return total;

}

}

طيب كل الحلول الى فوق دى حلول O(n) و ممكن نعمل حل يعتمد على معادله رياضيه يبقى فى الاخر سرعته O(1) اسرع بكتير من كل الى فوق دول فتعالى كده لو فكرنا براحه و قولنا تعالى نعمل معادله رياضيه هنعبر عن ال Number == 5 n لو هنبتدى نحسبها من شمال لليمين و من اليمين للشمال

من الشمال الاول ثم المعادله بالعكس ثم هنجمعهم على بعض

R= 1 + 2 + (n-2) +(n-1)+(n-0) = 1+2+3+4+5 = 25

R=(n-0) + (n-1) + (n-2) + 2 + 1 = 5+4+3+2+1 = 25

---------------------------------------------------------------------------

2R = (n+1) + (n+1) + (n+1) + (n+1) + (n+1) = 5 (n+1) = 50

طيب ناتج جمعهم طلع

2R = 5 \* (n+1)

طب ال 5 دى هى ال n اساسا اذاى هيصبح الناتج كده

2R = n \* (n+1)

و بما ان ده ناتج جمع المعادلتين على بعض يعنى و انا عاوز ناتج معادله واحده فالمفروض هقسم الناتج على 2

2R = (n \* (n+1)) / 2 = (5\*6( / 2 = 30/2 = 15

public class Program

{

public static void Main()

{

int Res = AddUpTo(5);

Console.WriteLine($"Res = {Res} .");

}

public static int AddUpTo( int number)

{

return (number \* (number+1)) /2;

}

}

اهه كده الكود ده O(1) اسرع بكتير جدا و اقل فى استهلاك الميمورى كمان

MB

##################################################################

Search & Sort Algorithms

-------------------------------

هى عباره عن خوارزميات لحل مشاكل الترتيب و البحث و دول و احنا بنتعملهم محتاجين نسأل بعض كذا سؤال   
1- الخوارزميه دى بتحل مشكله ايه ؟؟

2- اذاى هى شغاله و السودو كود بتعها   
3- نحللها نشوف بتاخد وقت اد ايه و بتستهلك من الميمورى ايه

4- الكود بقه

5- تحسين الكود

MB

##################################################################

Linear Search

-------------------

ده الجوريزم بيحل مشكله البحث عن شئ جوه اراى فيها العناصر مش مترتبه او مترتبه عن طريق انه بيمشى على العناصر عنصر عنصر واحد واحد لحد ما يوصل للعنصر المطلوب   
يعنى مثال انا عندى مجموعه ارقام جوه اراى ذى الى تحت دى و بدور جواها على رقم 22 فعيقعد يمشى جوه الاراى عنصر عنصر من الاول للاخر لحد ما يلاقى الى هو عاوزه ساعتها خلاص يقف و يرجع رقم ال index طب لو ملقاش العنصر ده بيرجع -1

A diagram of a graph

Description automatically generated

ال pseudo code : فور لوب بتمشى على كل العناصرمن اول انديكس 0 لحاد الانديكس الى اقل من n=arr.length لو العنصر الحالى بيساوى التارجت هيرجع الانديكس بتاع العنصر الحالى و لو ملقاش العنصر خالص جوه الاراى هيررجع -1 فنفهم ساعتها انه ملقاش العنصر الى بندور عليه جوه الاراى

A black background with white text

Description automatically generated

ال worst case انى الاقى الحاجه الى بدور عليها فى اخر الاراى يبقى كده هيلف على العناصر بتاعت الاراى كلها او لمقهوش اصلا فبرضوا هيمشى على كل الاراى يدور يبقى سرعته O(n)

ال average case انى الاقى الحاجه الى بدور عليها فى نص الاراى يبقى كده هيلف على نص عناصر بتاعت الاراى كلها يبقى سرعته O(n)

ال Best case انى الاقى الحاجه الى بدور عليها فى أول الاراى يبقى كده همش هيلف اصلا ده هيلاقه اول ما يقول سلام عليكم فكده سرعته O(1) سريع جدا

ألكود يا رايق بقه

public class Program

{

public static void Main()

{

int[] ints = {7,9,10,6,8,5,1,2,4,3};

int Res = LinearSearch(intArr: ints,target:6);

Console.WriteLine($"Target 6 is in index : {Res} .");

}

public static int LinearSearch(int[] intArr, int target)

{

for (int index = 0; index < intArr.Length; index++) // searching in all items

{

if (intArr[index] == target) // Comparing current item with target

{

return index; // if found return index

}

}

return -1; // if not found return -1

}

}

MB

##################################################################

Binary Search

-----------------------------------------------------