تحليل الخوارزميات
المحاضرة الاولى

Abdalmohaymen alesmaeel
الجامعة الدولية للعلوم والنهضة

محاور المحاضرة

- الهدف من المقرر
- ماهي بنى المعطيات
- الهدف والفائدة من بنى المعطيات
- ماهي الخوارزمية
- مكونات الخوارزمية برمجيا
- تحليل المشكلة وحلها
- ماذا يعني تحليل الخوارزمية
- لماذا نحلل الخوارزمية
- برهان صحة الخوارزمية في حل المشكلة
- كيف نقيس جودة الخوارزميات المطبقة
- الدراسات التجريبية
- سلبيات الدراسات التجريبية
- التعقيد
- كيف نقوم بحساب التعقيد الخوارزميات
- ماهي مستويات التعقيد

الهدف من المقرر

- التعرف على بنى المعطيات وطرق إنجازها
- عرض بعض الخوارزميات الأساسية
- تمكين الطالب من تحليل وتطوير الخوارزمية
- من ثم استخدام هذه الخوارزميات في التطبيقات من خلال أمثلة عملية
- بالتالي القدرة على حل المسائل والمشاكل
- **الجزء التطبيقي: لغة البرمجة جافا**

ماهي بنى المعطيات

- **بنى المعطيات (Data Structures):** هي طرق منظمة لتخزين وتنظيم البيانات في **الذاكرة الحاسوبية** بحيث يمكن إدارتها والوصول إليها بفعالية. الهدف من استخدام بنى المعطيات هو تحقيق الكفاءة في التنفيذ والاستفادة المثلى من موارد النظام.
- أنواع بنى المعطيات
  - بنى المعطيات الأساسية
    - جميع أنواع البيانات هي بالنهاية تعتبر بنى معطيات يتم تخزين البيانات فيها
  - بنى المعطيات الخطية Linear Data Structures
    - المصفوفات
    - القوائم والقوائم المترابطة (اللوائح)
    - المكدسات والأرتال
  - بنى المعطيات الغير خطية Non-Linear Data Structures
    - جداول التقطيع
    - الأشجار
    - البيان

الهدف والفائدة من بنى المعطيات

- لنتخيل عند عملية تسجيل الدخول الى احدى منصات شبكات التواصل الاجتماعي
- مليارات الحسابات (كيف تتم المقارنة)
- تتطلب مثل هذه البرمجيات طرق وأدوات تحقق أداء عالي
- مثال:
- لدينا 1000 موظف ونريد البحث عن الموظف خالد (وكل عملية بحث لنفترض تتطلب 1 ثانية)
- باستخدام البحث الخطي فإننا بحاجة الى 1000 ثانية .
- لكن ماذا لو كان البحث وفق فهارس المحارف الأبجدية بالتالي للوصول للموظف خالد ربما يتطلب عمليات اقل بكثير بالتالي فإن الفوائد هي:

1) تحسن كفاءة استخدام الموارد
2) تسريع عمليات البحث والاضافة والتعديل
3) تحسين أداء العمليات بشكل عام
4) تجربة مستخدم جيدة للعملاء من خلال استجابة تنفيذ سريعة

ماهي الخوارزمية

- الخوارزمية هي مجموعة من التعليمات المنظمة والمحددة جيدًا التي تقوم بتنفيذ مجموعة من الخطوات لتحقيق مهمة معينة **هدف** أو حل مشكلة.
- في العلوم الحاسوبية والرياضيات: الخوارزميات هي الأساس لمعظم العمليات الحسابية والبرمجية. يتم تصميم الخوارزميات لتعمل على البيانات، حيث تقوم بمعالجتها وتحليلها وإنتاج نتائج بناءً على هذه البيانات. يجب أن تكون الخوارزمية:

1. واضحة وغير مبهمة
2. لها نقطة بداية ونقطة نهاية
3. فعّالة
4. قابلة للتنفيذ

مكونات الخوارزمية برمجيات

- مدخلات: بيانات يتم إدخالها في البرنامج.
- مخرجات: بيانات ينتجها البرنامج.
- خطوات تنفيذية: تعليمات مرتبة لتنفيذ مهمة.
  - متغيرات: تخزين البيانات لاستخدامها لاحقًا.
  - تعابير: جمل تحسب قيم من المتغيرات.
  - شروط: اختبار تنفيذ الكود بناءً على حالة.
  - حلقات: تكرار كود حتى تحقق شرط.
  - مصفوفات: تخزين متعدد القيم في متغير واحد.
  - توابع: كتلة كود تنفذ مهمة محددة.
- تعليقات والتوثيق: شرح الكود للمطورين.

تحليل المشكلة وحلها باستخدام الخوارزمية

- **فهم المشكلة:** تحديد الهدف ومتطلبات المشكلة بوضوح.
- **تحديد المدخلات:** ما هي البيانات الداخلة؟ حجمها ونوعها.
- **تحديد المخرجات:** ما هي النتائج المتوقعة؟ كيف سيتم تقديمها؟
- **تصميم الخوارزمية:** وضع خطوات متسلسلة وواضحة لحل المشكلة.
- **كتابة البيانات الوهمية:** تحديد أمثلة بيانات لاختبار الخوارزمية.
- **تنفيذ الخوارزمية:** كتابة الكود بناءً على الخطوات المصممة.
- **التحقق من الخوارزمية:** استخدام البيانات الوهمية للتأكد من صحة الحل.
- **تحليل الخوارزمية:** تقييم أداء الخوارزمية من حيث الزمن والمساحة.
- **تحسين الخوارزمية:** البحث عن طرق لجعل الخوارزمية أكثر كفاءة.
- **التوثيق:** كتابة تعليقات وتوثيق لشرح الخوارزمية وكيفية عملها.

ماذا يعني تحليل الخوارزمية ؟

1. **صحة الخوارزمية:** يتعلق هذا الجانب بالتحقق من أن الخوارزمية تقوم بحل المشكلة المطروحة بشكل صحيح. هذا يعني أنه لكل مجموعة من المدخلات، يجب أن تنتج الخوارزمية المخرجات المتوقعة والصحيحة.
2. **كم العمل اللازم للوصول إلى الحل:** يشير هذا إلى تقدير كمية العمليات الحسابية أو الخطوات التي تتطلبها الخوارزمية للوصول إلى حل المشكلة. هذا التقدير مهم لفهم كفاءة الخوارزمية وكيفية أدائها مع تزايد حجم المدخلات.
3. **حجم الذاكرة الضروري:** يتناول هذا الجانب كمية الذاكرة أو المساحة التخزينية المطلوبة لتنفيذ الخوارزمية. يعتبر تقليل استهلاك الذاكرة مهمًا لزيادة كفاءة الخوارزمية، خصوصًا في الأنظمة ذات الموارد المحدودة.
4. **البساطة:** يشير إلى مدى سهولة فهم الخوارزمية وتنفيذها. الخوارزميات البسيطة أسهل في التحليل، التصحيح، والصيانة. بالإضافة إلى ذلك، تميل البساطة إلى تقليل احتمالية الأخطاء.
5. **الأمثلية:** تقييم مدى كفاءة الخوارزمية مقارنةً بالحلول الأخرى الممكنة. الخوارزمية المثالية هي التي تحقق أفضل أداء ممكن (أقل زمن تنفيذ وأقل استهلاك للموارد) لحل مشكلة معينة.

لماذا نحلّل الخوارزمية

- تحسين الخوارزمية
- إجراء مقارنة بين مجموعة من الخوارزميات التي تحل المسألة نفسها بهدف اختيار الأنسب

برهان صحة الخوارزمية

- الشروط المسبقة Preconditions : يجب التحقق من طبيعة الدخل الذي تطبق عليه الخوارزمية
- الشروط اللاحقة Postconditions : يجب ان نعرف طبيعة الخرج الذي يجب أن تولده الخوارزمية
- برهان صحة الخوارزمية يعني برهان التالي:

"في حال كانت الشروط المسبقة محققة و إذا انتهى تنفيذ الخوارزمية وكانت الشروط اللاحقة محققة عندها يعتبر بناء الخوارزمية صحيحا"

ما العمل اللازم للوصول إلى الحل

- زمن تنفيذ الخوارزمية على آلة ما ؟
- ما هو تعقيد الخوارزمية ؟
- كيف نقيس تعقيد الخوارزمية ؟
  - دراسة تجريبية
  - التعقيد الزمني والمكاني

قياس زمن التنفيذ

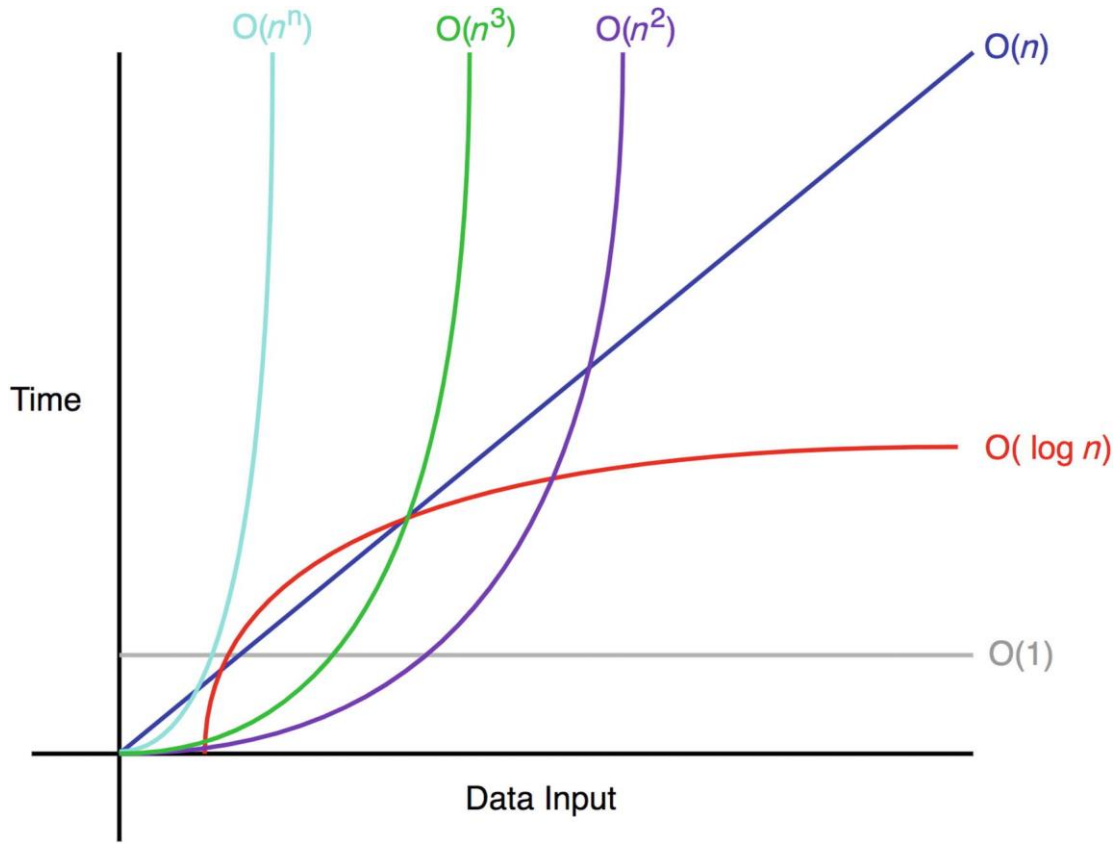
- دراسة تجريبية Experimental Study
  - اكتب برنامج ينجز الخوارزمية
  - نفذ البرنامج على مجموعة من المعطيات تختلف في حجمها وفي تركيبها
  - استخدم طريقة برمجية للحصول على قياس دقيق لزمن التنفيذ الفعلي عبر قياس الزمن في بداية التنفيذ وفي نهاية التنفيذ وحساب زمن التنفيذ
  - اذا كان هناك عددا كبيرا من مجموعات المعطيات فبإمكانك الحصول على منحن يوافق تعقيد الخوارزمية .

سلبيات الدراسات التجريبية

- للدراسات التجريبية عدة مساوئ:
  - من الضروري تنفيذ واختبار الخوارزمية لتحديد زمن تنفيذها.
  - يمكن إجراء التجارب على مجموعة محدودة فقط من الدخل, ويمكن ان تكون هذه المجموعات غير ذات دلالة على زمن تنفيذ مدخلات أخرى غير موجودة ضمن التجربة .
  - تباين البيئات التجريبية: النتائج يمكن أن تتأثر بشدة بالبيئة التجريبية، بما في ذلك الأجهزة المستخدمة وإعدادات النظام، مما يصعب تعميم النتائج.
  - تحديات التكرار: قد يكون تكرار الدراسات التجريبية بنفس النتائج صعبًا بسبب الاختلافات في البيئات التجريبية وإعدادات التنفيذ.
  - محدودية النطاق: الدراسات التجريبية تعتمد على السيناريوهات المختارة للتجربة، وقد لا تغطي جميع الحالات الاستخدامية الممكنة للخوارزمية، مما يحد من قدرتها على التنبؤ بالأداء في جميع الظروف.
  - عند مقارنة خوارزميتين , يجب استخدام نفس البيئة البرمجية software والعتادية hardware .

التعقيد الزماني

- هو تقدير لعدد الخطوات او العمليات الأساسية التي تقوم بها الخوارزمية ونركز هنا على كلمة تقدير حيث هنا يتم حساب عدد العمليات التي سيتم اجراؤها عند تنفيذ الخوارزمية وبالتالي لا حاجة الى تجريب الخوارزمية أبدا وقياس أدائها من خلال دراسة التعقيد الخاص بها



التعقيد

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	Θ(1)	Θ(n)	Θ(n)	Θ(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Queue	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	Θ(n)	Θ(n)	Θ(1)	Θ(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	N/A	Θ(1)	Θ(1)	Θ(1)	N/A	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(n)	O(n)	O(n)	O(n)
B-Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	N/A	Θ(log(n))	Θ(log(n))	Θ(log(n))	N/A	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
KD Tree	Θ(log(n))	Θ(log(n))	Θ(log(n))	Θ(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)

التعقيد Complexity

- منهجية عامة لتحليل زمن تنفيذ الخوارزميات :
  - تستخدم وصفاً عالي المستوى للخوارزمية بدلاً من اختبار أحد تنفيذاتها.
  - تأخذ بعين الاعتبار جميع المدخلات المحتملة.
  - تسمح بتقييم فعالية أي خوارزمية بطريقة مستقلة عن البيئة البرمجية والعتادية

تعقيد الخوارزميات

- العمليات الأولية: حسابات منخفضة المستوى مستقلة إلى حد كبير عن لغة البرمجة ويمكن تعريفها بالرماز الوهمي او المفترض , مثل :
  - استدعاء طريقة والعودة من طريقة عبر return
  - اجراء عملية حسابية (مثل الجمع ,.....)
  - مقارنة عددين , الخ
- يمكننا بالتدقيق في الرماز المفترض عد العمليات الأولية التي تنفذها الخوارزمية

التدوين المقارب

The top complexity time classes are:

$O(1)$: Constant time
$O(n)$: Linear time
$O(n^2)$: Quadratic time
$O(n^c)$: Polynomial time (general case c>=2)
$O(\log_b n)$: Logarithmic time
$O(n \log_b n)$: Linearithmic time
$O(2^n)$: Exponential time
$O(n!)$: Factorial time

كيف نقوم بحساب التعقيد الخوارزميات

عدد عمليات الاسناد

عدد العمليات الحسابية

عدد عمليات المقارنة

من اجل الحلقات:

- نعدّ عدد مرات الدخول في الحلقة
- يمكن ان نختار عمليات أساسية ذات علاقة وثيقة بالعمل الذي تقوم به الخوارزمية ونعدّها فقط, دون الالتفات إلى العمليات الثانوية

أمثلة على عمليات أساسية

المسألة	العمليات الأساسية
البحث عن الكلمة x داخل صفيفة من الأسماء	مقارنة كلمتين
جداء مصفوفتين	جداء عددين حقيقين
فرز صفيفة من الأعداد الحقيقية	مقارنة عددين من صفيفة الدخل
مسألة تستدعي إجرائيات عودية	عدد مرات استدعاء الإجرائية العودية

قواعد حساب زمن تنفيذ الخوارزمية

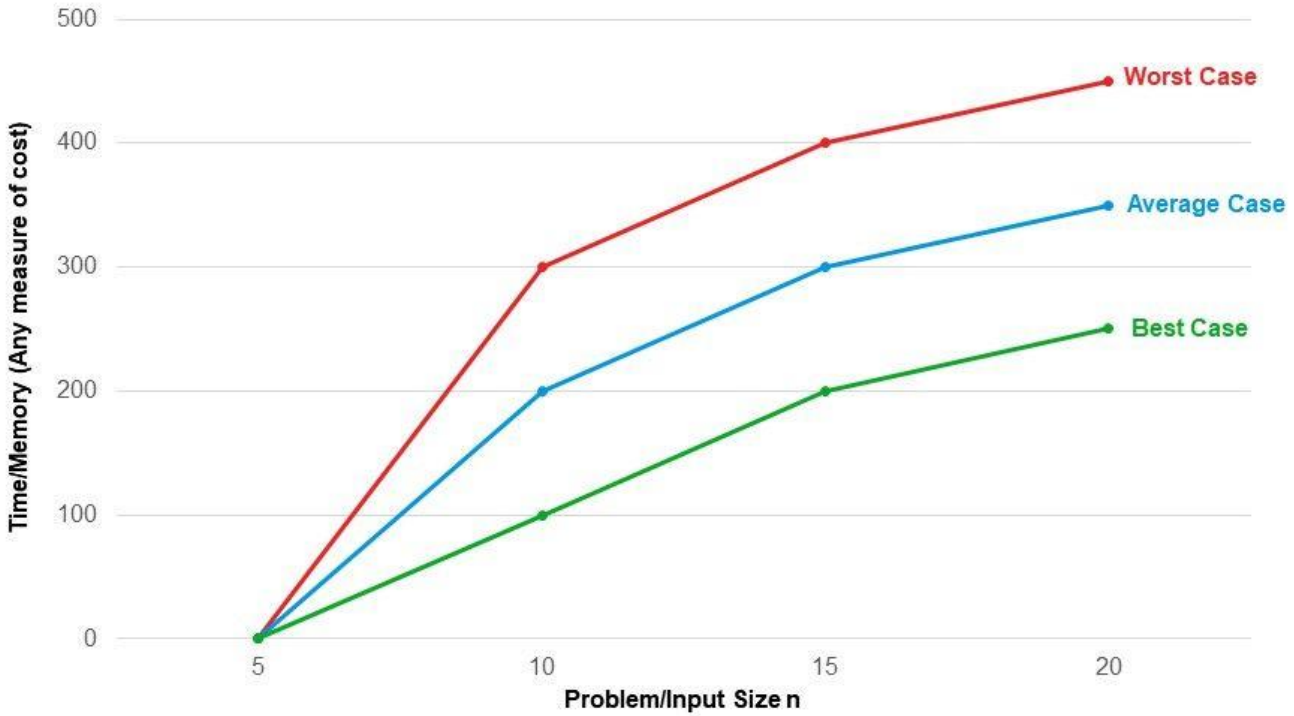
- بعد تحديد أنواع العمليات الأولية , يتم عد العمليات من كل نوع , وذلك تبعاً لما يلي:
  - بالنسبة إلى العمليات المتتالية فإن الزمن هو مجموعة من الأزمنة
  - بالنسبة الى العملية الشرطية
- X=if(c) then (a) else (b)

يحسب حداً أعلى لعدد العمليات اذا كان T(x) عدد العمليات الأولية في X فإن :

- $T(x) \approx T(c) + Max(T(a),T(b))$
  - بالنسبة إلى الحلقات, فإن عدد العمليات هو :
- $\sum T(i)$
  - حيث i هو المتحول الذي يتحكم بالحلقة, و T(i) عدد العمليات الأولية المتعلقة بالمرور رقم i في الحلقة
  - بالنسبة إلى طلب (استدعاء) إجرائية تحسب درجة تعقيد الإجرائية باتباع نفس القواعد (حساب عدد العمليات الأولية في الإجرائية
  - بالنسبة إلى الإجرائيات العودية , تستخدم علاقتها العودية لحساب درجة التعقيد
- T(n)=f(T(g(n)))

مستويات التعقيد

Graphical Representation of Best Average And Worst Case



This graph/chart is linked to excel, and changes automatically based on data. Just left click on it and select "Edit Data".

- الحالة المثلى ليست الحالة الاعتيادية في الطبيعة
- نقوم باتخاذ قرار حول الحالة التي نريدها بحسب البرنامج فمثلا:
  - في البرامج التي تعتمد مضادات الطيران او القطاعات الصحية يجب الاخذ بأسوء حالة
  - بينما لا مشكلة في برامج أخرى
- يمكن ان يكون تنفيذ خوارزمية ما على مجموعة معينة من المعطيات أسرع من تنفيذها على معطيات أخرى بحسب أنواع المعطيات وحجمها
  - مثال : فرز مصفوفة شبه مرتبة بالمقارنة مع فرز مصفوفة غير مرتبة
- يمكن ان يكون إيجاد الحالة الوسطية صعبا جدا لذلك يقاس تعقيد الخوارزميات عادة بالتعقيد في الحالة الأسوء
- كذلك تكون معرفة التعقيد الزمني في الحالة الأسوء في بعض التطبيقات الخاصة أمراً أساسيا
- التحكم في المطارات
- العمليات الجراحية

التعقيد في أسوء الحالات Worst Case Complexity

- تابع التعقيد في أسوء الحالات W(n) هو أكبر عدد من العمليات التي يمكن أن تقوم به الخوارزمية لحل مسألة من أجل أي دخل حجمه n .
- $W(n)=\max\{t(i): i \in Dn$
- حيث
  - D_n مجموعة معطيات الدخل التي حجمها n،
  - i أحد عناصر D_n، و t(i) عدد العمليات الأساسية التي تنفذها الخوارزمية لحل المسألة من أجل الدخل i.

مثال: البحث عن عدد داخل صفيفة من الأعداد

Algorithm seqSearch(A, k)

Input: An integer array A, an integer k
Output: The smallest index i with A[i] = k,
if such an i exists, or -1 otherwise.

```
1. for i ← 1 to A.length do
2.     if A[i] = k then
3.         return i
4. return -1
```

ما الخرج والدخل؟
ما العملية الأساسية؟
ما حجم المعطيات؟
ما التعقيد؟
ما هي أسوأ الحالات؟
ماذا لو عكسنا اتجاه حلقة for؟

التعقيد في أسوء الحالات

حالة ألا ينتمي العدد إلى المصفوفة

حالة العدد موجود في آخر موقع

ملاحظة أسوء حالة مرتبطة بطريقة عمل المسألة

لنتخيل خوارزمية تعمل بطريقة البحث من الخلف الى الامام فانه يختلف أسوء حالة ماهي

Algorithms Websites

- https://visualgo.net/en
- https://codeforces.com/
- https://leetcode.com/
- https://www.hackerrank.com/
- https://projecteuler.net/
- https://www.codewars.com/
- https://codesignal.com/