



جامعة دمشق

كلية الهندسة المعلوماتية

السنة الثانية

الخوارزميات وبني المعطيات /1/

## الوظيفة الفصلية

### المشروع /2/

تقديم الطلاب:

حمزة سمير عمّار

رائد محمد زهير السبيناتي

راما معين ربحاوي

رغد عامر الحلبي

عبد الرحمن محمد بشار الأرمشي

إشراف: م. سعيد سريحي

الاثنين 6/1/2020

## المسألة الأولى:

في البداية، تعاملنا مع بنية struct information تحوي على (الاسم الكامل للشخص، العمر).

تم استخدام بنية trie للتعامل مع الأرقام الخاصة للأشخاص بالسجلات المدنية (id)، تحوي مصفوفة مؤشرات، كما تحوي مؤشر من نوع البنية السابقة (information)، حيث يتم بواسطتها الربط بين الـ id ومعلومات الشخص، وبهذا يصبح المؤشر الموجود في بنية trie مشابه لعمل المتحول البوليفاني (بحيث كان المؤشر يحوي على معلومات الشخص فهذا دليل على وجود شخص يحمل هذا الـ id، وإلا فإنه سيشير إلى NULL)

التوابع المستخدمة:

(1) تابع الإضافة (add): يقوم بالربط بين مؤشر information والـ id، أي أننا نقوم فعلياً ببناء مخطط لبنية trie (حجز بيئة trie لأجل كل خانة من خانات الـ id).

**تعقيد تابع الإضافة يساوي: عدد الخانات لأكبر id مُدخل.**

**أي:  $Complexity: O[\log_{10} (ID\_MAX)]$**

(2) تابع البحث (ser): يقوم بالمرور على مخطط بنية الـ trie كاملاً، في حال العثور على الـ id الذي نبحث عنه، يقوم بإرجاع مؤشر على معلومات الشخص صاحب الـ id المدخل.

**تعقيد تابع البحث يساوي: عدد الخانات لأكبر id مُدخل.**

**أي:  $Complexity: O[\log_{10} (ID\_MAX)]$**

**فيكون تعقيد الخوارزمية كاملةً هو  $Complexity: O[Q \cdot \log_{10} (ID\_MAX)]$**

## المسألة الثانية:

- بنى المعطيات المستخدمة:

`vector< vector<string> > row,col`: لتخزين كلمات الأسطر (row) وكلمات الأعمدة (col).

`vector< vector<char> > cur`: لتمثيل الرقعة (حيث أن الرقعة تحوي القيم الابتدائية '#').

- خوارزمية الحل:

نقوم من خلال تابع `backtrack` بالمرور على الأسطر، حيث أنه في كل سطر نقوم بتجريب كافة الخيارات (الكلمات) المتاحة من أجل هذا السطر.

وبالتالي نقوم بتسجيل الحل في مصفوفة الـ `cur`، ثم نقوم بالاستدعاء من أجل السطر الذي يليه. عند المرور على كافة الأسطر، نستدعي تابع الفحص (`check`) الذي يقوم بالمقارنة بين كل عمود من الأعمدة الناتجة عن تجريب كلمات الأسطر في تابع الـ `backtrack`.

وعند معرفة الحل الناتج نستدعي تابع الطباعة (`Print`)، الذي يقوم بطباعة الرقعة الناتجة عن تابع الـ `backtrack`.

**التعقيد = (عدد الأسطر) عدد الكلمات بالسطر الواحد  $k_{max}$**

**أي: Complexity:  $O[n^{k_{max}}]$**