

الباب التاسع

البرمجة الديناميكية

معضلة الحقيبة:

لديك حقيبة تستطيع تحمل 4 كجم، ونريد أن نملأها بمنتجات تحتوي على أعلى قيمة ممكنة ولدينا ثلاث منتجات لنختار بينهم:

1. راديو - وزن = 4 كجم، سعر = \$3000

2. لايتوب - وزن = 3 كجم، سعر = \$2000

3. جيتار - وزن = 1 كجم، سعر = \$1500

يمكننا أن نجد الحل عن طريق إيجاد كل الاحتمالات الممكنة ثم إيجاد الأفضل بينهم، ولكن هذا يستهلك الكثير من الوقت $O(2^n)$ يمكننا أيضا إيجاد الإجابة باستخدام الgreedy algorithm ولكنها إجابة تقريبية، لإيجاد الحل نحتاج إلى استخدام البرمجة الديناميكية.

البرمجة الديناميكية:

تعتمد البرمجة الديناميكية على حل مشاكل فرعية وتراكمهم حتى نصل لحل المشكلة الأصلية، سنبدأ حل المشكلة السابقة بإيجاد الحل لحقائب أصغر، البرمجة الديناميكية دائما تحتاج إلى جدول لإيجاد الحل

COLUMNS ARE KNAPSACK SIZES FROM 1 TO 4 POUNDS

	1	2	3	4
GUITAR				
STEREO				
LAPTOP				

ONE ROW FOR EACH ITEM TO CHOOSE FROM

لنبدأ بالصف الأول، نحن نريد الإجابة على السؤال التالي: هل نأخذ الجيتار أم لا؟ لنجيب عليه يجب أن نعرف إذا كانت الحقائب تتحمل الجيتار أم لا، إذا كانت الإجابة لا نضع صفر في الخلية وإذا كانت نعم نضع سعر الجيتار (للصف الأول فقط)

	1	2	3	4
GUITAR	\$1500 G	\$1500 G	\$1500 G	\$1500 G
STEREO				
LAPTOP				

في الصف الثاني نريد الجواب على سؤال آخر، هل نأخذ الراديو، أم الجيتار، أم كلاهما؟ للإجابة على هذا السؤال يجب أن نجيب على أسئلة أخرى أولا. هل الحقيبة تتحمل الراديو؟ إذا كانت الإجابة لا نضع في الخلية نفس القيمة من الصف السابق، وإذا كانت الإجابة نعم، ولكن لا يوجد أي مساحة إضافية نقارن بين قيمة الراديو والقيمة السابقة للحقيبة ونضع الأكبر بينهما.

	1	2	3	4
GUITAR	\$1500 ↓ G	\$1500 ↓ G	\$1500 ↓ G	\$1500 ↓ G
STEREO	\$1500 G	\$1500 G	\$1500 G	\$3000 S
LAPTOP				

ولكن ماذا إذا كانت الحقيقة تكفي، ولكن لا يزال هناك مساحة؟ نأخذ سعر المنتج ونجمعه على القيمة الموجودة في الحقيقة التي تحتوي على نفس حجم المساحة الإضافية في الصف السابق ونقارنه بالقيمة السابقة للحقيقة الحالية.

	1	2	3	4
GUITAR	\$1500 ↓ G	\$1500 ↓ G	\$1500 ↓ G	\$1500 ↓ G
STEREO	\$1500 G	\$1500 G	\$1500 G	\$3000 S
LAPTOP	\$1500 G	\$1500 G	\$2000 L	\$3500 LG

↑
THE ANSWER!

ملحوظات:

- إذا أضفنا أي منتجات آخر للجدول فالقيمة النهائية قد تتغير وقد تبقى كما هي.
- ترتيب الصفوف لا يؤثر في النتيجة النهائية
- إذا وضعت المشتريات عند العواميد فالأمر لن يسبب مشكلة في هذه المعضلة، ولكن قد يحدث تأثير في معضلات أخرى
- إذا أضفنا منتج ذو وزن أصغر (نصف كجم) يجب إضافة عواميد لحقائب ذات أوزان تناسبه (0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4)
- البرمجة الديناميكية لا تعمل مع أنصاف الحلول إما أن تأخذ الشيء أو تتركه لا يمكنك أن تأخذ جزءا منه (نصف كجم أرز من حقيبة مليئة به).
- البرمجة الديناميكية تعمل فقط حين تكون المشاكل الفرعية مستقلة ولا تؤثر على بعضها البعض
- في المعضلة السابقة سيكون الحل دائما مجموع حقيبتين كحد أقصى، ولكن كل حقيبة منهما قد تحتوي على حقائب أخرى
- ليس من الضروري أن تمتلئ الحقيبة بالكامل
- تحاول البرمجة الديناميكية الوصول لأفضل حل تحت شروط معينة

حين تخطئ إملائيًا كيف يعرف الكمبيوتر الكلمة المقصودة؟ يقوم بمقارنة الخطأ بآلاف الكلمات عنده عن طريق البرمجة الديناميكية بهدف العثور على كلمة تحتوي على أكبر قدر من الحروف المتشابهة

ستضع حروف الكلمتين في جدول وتقارن بينهما، إذا كان الحرفين المتقابلين مختلفين نضع 0 أما إذا كان متماثلين نضع 1 إذا كنا في الصف الأول وإذا لم نكن نأخذ القيمة الموجودة بالأعلى يسارا ونضيف عليها 1.