

MONEY SYSTEMS

Δυναμικός Προγραμματισμός

Για κάθε τιμή του V υπολογίζουμε από για προηγούμενες τιμές (0 μέχρι $V-1$) πόσες λύσεις υπάρχουν. Αν $c[i]$ η αξία του κέρματος i και $V-c[i] \geq 0$ τότε μπορείς να κάνεις την αξία V με όσους τρόπους έκανες την $V-c[i]$ προσθέτοντας το κέρμα $c[i]$. Έχοντας σαν αρχικό condition ότι μπορούμε να κάνουμε την αξία 0 με 1 τρόπο (κανένα κέρμα) υπολογίζουμε από τα κάτω προς τα πάνω τις τιμές του V .

Μια άλλη καλή εξήγηση είναι η εξής: αν $c(n,k)$ οι τρόποι που μπορείς να κάνεις την αξία n χρησιμοποιώντας τα πρώτα k νομίσματα τότε:

$$c(n,k) = c(n,k-1), c(n-c[k],k)$$

δηλαδή οι τρόποι που μπορείς να κάνεις την αξία n χρησιμοποιώντας τα πρώτα k νομίσματα είναι οι τρόποι που μπορείς να κάνεις την αξία n χρησιμοποιώντας τα πρώτα $k-1$ νομίσματα (δηλαδή χωρίς να επιλέγεις το νόμισμα k + οι τρόποι που μπορείς να κάνεις την αξία $n-c[k]$ χρησιμοποιώντας τα πρώτα k νομίσματα (χρησιμοποιώντας δηλαδή το νόμισμα k). Η πρώτη επεξήγηση είναι η παρόμοια απλά πιο απλοποιημένη. (βρίσκουμε την απάντηση για κάθε V και μετά προχωρούμε στο $V+1$ προσθέτοντας στο ίδιο array αντί να αλλάζουμε γραμμή.)

Οι μαθητές εκτός της θεωρίας της usaco για το dp είναι καλό να δουν το ακόλουθο tutorial:

<https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/>