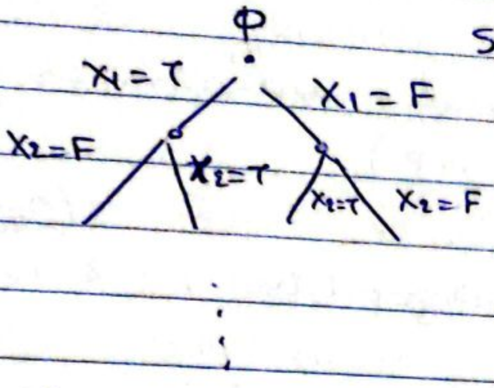
Δε 8/1/2024

SAT



Μια λύση επαληθεύεται σε πολυωνυμικό χρόνο

Για κάθε γλώσσα της NP έχω μια σχέση R για την οποία μπορώ να ελέγξω σε πολ. χρόνο αν xRy και $|y| \le p(|x|)$ για p: πολυώνυμο

Αν έχω σχέση $R_{SAT}(\phi, y)$: η y είναι ικανοποιούσα ανάθεση της φ, $y = a_1 a_2 \dots a_n$, $a_i \in \{T, F\}$

$$\phi|_{x_1 = a_1, \dots, x_n = a_n} = T$$

και ικανοποίηση

- $|y| \le |\phi|$
- R_{SAT}: πολυωνυμ. αποκρ. γιατί έχω αντικατάσταση και εύρεση αληθοτιμής της φ

x∈L μπορώ να φτιάξω δυαδικό δέντρο με ύψος το πολύ $p(|x|)$

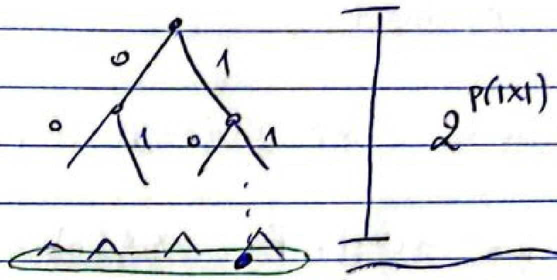
$R_x(x, y)$: y = κλάδος υπολογισμού της M με είσοδο x που απαντάει ναι

Γιατί από το $L = \{x \in \Sigma^* : \exists y \in \Sigma^*, (x,y) \in R\}$ μπορώ να φτιάξω μια NDTM?

Αλγ.

(Δοσμένης της σχέσης άρα ζ της L)
- πώς ξέρω ότι οι λύσεις είναι πολυωνυμ. στο μήκος της εισόδου;
- ότι η λύση είναι σε πολυωνυμικό χρόνο ναι (ή όχι);

Μια λύση έχει (έστω) δυαδική αναπαρ. Αφού αυτή έχει πολυωνυμ. μέγεθος σε κάθε επίπεδο διαλέγω ψηφίο για τη λύση



Σε αυτό το επίπεδο → τρέχω $A(x,y)$ (μπορώ να παραβλέψω τα μικρότερα πιστοποιητικά με μια αλλαγή)

- Το συμπληρωματικό πρόβλημα έχει τα συμπλήρωμα που δίνουν όχι αντί για ναι

- $\overline{SAT} = UNSAT = \{ \phi : \neg\phi :$ ταυτολογία$\}$
$\in$ coNP ζ μάλιστα είναι coNP-πλήρες
(δεν ξέρω αν είναι γνήσιο)
$P \subseteq coNP \cap NP$

* τα προβλήματα βελτιστοποίησης δεν ανήκουν στο NP. το NP έχει προβλήματα απόφασης. Για αυτό τα μετατρέπουμε

* Θ. Cook

* Το $\phi \in SAT$ συνήθως το βλέπουμε σε CNF μορφή

(Τα μαθαίνουμε απ' έξω!!)

* 3-SAT : NP-πλήρες

ΔΙΑΦ 21 * 3-SAT(3) εμφάνιση κάθε μεταβλητής ≤ 3 φορές
NP-πλήρες

Αν έχω 3 καταφατικές εμφανίσεις $\rightarrow$ ικανοποιώ τα clauses στα οποία εμφανίζεται $\rightarrow$ απασχολούμαι με τα άλλα

ΔΙΑΦ 22 * MAX2SAT NP-πλήρες

ΔΙΑΦ 23 * MAX IND SET NP-πλήρες (μέγιστο $\rightsquigarrow \geq k$)

$a(\varphi)=1$ για $a(x_1)=F$ $a(x_2)=a(x_3)=T$

ΔΙΑΦ 25 * MIS (4) NP πλήρες (γράφημα μέγιστου βαθμού 4)

ΔΙΑΦ 26 * Vertex Cover, Ind Set, Clique

* Min VC $\leq_p$ Ακερ. Γρ. Προγ (ILP)

$$\min \sum_{v \in V} x_v$$
s.t.

$$x_v + x_u \geq 1 \quad \forall e=\{v,u\} \in E$$
και $x_v \in \{0,1\}$ $\forall v \in V$

ΔΙΑΦ 27, 28 * Set Cover

Υποδείξεις ασκήσεων 2ης προόδ.

(1η) $p_1 \ldots p_N$ Θέλω αλγόριθμο για μεγιστοποίηση κέρδους $\sum_{i=1}^{M} [p(s_i) - p(b_i)]$

DP

• κάθε μέρα μπορώ είτε να αγοράσω, είτε να πουλήσω

Έστω μια μέρα i
- Αν έχουμε αγοράσει → μπορούμε να πουλήσουμε
- Αν δεν έχουμε → να αγοράσουμε

ή - Να πουλήσω σοκολάτα που έχω αγοράσει πιο πριν (να βρω πότε)
- Να μην κάνω τίποτα
- Να αγοράσω ~~(αλλά πρέπει να βρω που θα~~

μέρα i
Profit [i,λ] : βέλτιστο κέρδος μέχρι μέρα i για λ αγορανή
Πρέπει να θυμάμαι πόσες αγορανή έχω κάνει

$$= \max \begin{cases} \text{Profit}[i-1, \lambda] \\ \cancel{\text{Profit}[} \quad \max_{1 \le j \le i-2} \left\{ \text{prof}[j, \lambda-1] + \underbrace{p(i)}_{\text{πουλώ } i} - \underbrace{p(j+1)}_{\text{την αγοράζω την j+1 (αποφασίζω)}} \right\} \end{cases}$$

Πρέπει να φτιάξω ξ αρχικές συνθήκες για να έχει νόημα

Πολυπλοκότητα : $N^2 K$ → 60%

καλύτερο max (ενημέρωση κάθε φορά)

όμως $= p(i) + \max_j \{ \text{prof}(j, \lambda-1) - P(j+1) \}$ NK

2η · Έστω ένα κλαδί του δέντρου

$$\underbrace{i - i\text{-}1 - i\text{-}2 \cdots 1 - 0}_{d(i)} \longrightarrow p(i) + s(i) \cdot d(i)$$

80%

DP $cost(i) = p(i) + \min_{j} \left\{ s(i)\left(d(i)-d(j)\right) + cost(j) \right\}$

πού θα κάνω στάση
Από εκεί η νέα
τη βέλτιστη λύση
της στάσης
αν
σώσει
σ' άλλον

$= p(i) + s(i)\cdot d(i) + \min_{0\le j<i} \left\{ \underbrace{cost(j)} - \underbrace{d(j)}\cdot s(i) \right\}$

αυξάνουν όσο αυξάνει το j και είναι της μορφής

$c_j - d_j\, x$ όπου x ελεύθερο

$c_1 < c_2 < c_3 \ldots$

$d_1 < d_2 < \ldots$

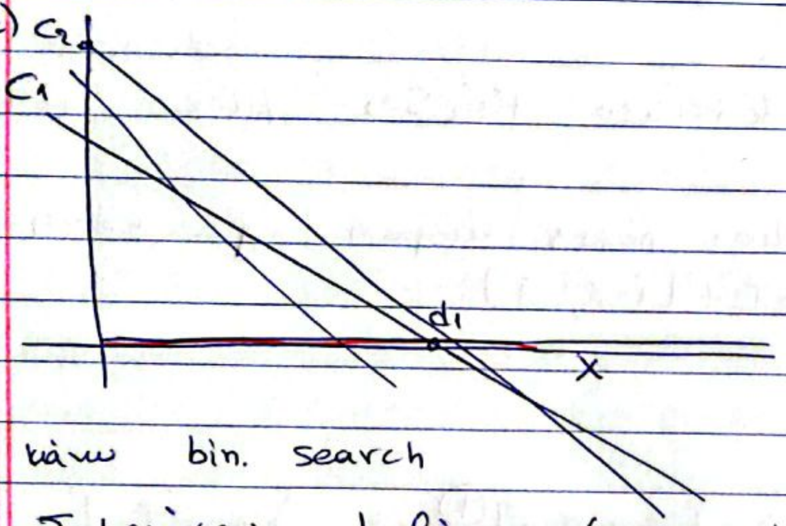
Πώς μπορώ μόνο με νέο x να υπολογίσω

$c_j - d_j \cdot x$

εύκολα;

110%
↓
σύνολο για δέντρο (persistent) (άλλος κώδικας)

Μοιάζουν έτσι



κάνω bin. search

Βλ. Σημειώσεις helios (συμπληρωμ. υλικό) convex hull trick
↳ φοιτητή