Армийн бодлогын Brute Force алгоритмын цагийн нарийвчлалын баталгаа

Алгоритмын зохиомж ба шинжилгээ

2026-02-11

1 Бодлогын тодорхойлолт

N цэрэг зүүн (L) ба баруун (R) гэсэн хоёр фронтод хуваагдсан. Цэрэг бүр $k = 15$ чадвараас дэд олонлог эзэмшсэн. Командлагч чадварын дугаар илгээхэд тухайн чадварыг эзэмшсэн бүх цэрэг фронтоо солино. Командыг хэдэн ч удаа илгээж болох ба зүүн фронт дахь цэргүүдийн тоог хамгийн ихэсгэнэ.

2 Гол ажиглалт

Лемм 1: Тэгш тоогоор илгээх нь ялгаагүй
Нэг чадварын дугаарыг m удаа илгээсэн гэвэл:
- m тэгш бол: нөлөө байхгүй (анхны байдалдаа буцна)
- m сондгой бол: нэг удаа илгээсэнтэй адил

Тиймээс чадвар бүрийг 0 эсвэл 1 удаа илгээхэд хангалттай.

Баталгаа:
Цэргийн фронт солилт нь toggle үйлдэл юм. Нэг чадварыг m удаа toggle хийвэл эцсийн төлөв нь $m \bmod 2$-с хамаарна. Хэрэв $m \equiv 0 \pmod 2$ бол анхны төлөвтөө буцах ба $m \equiv 1 \pmod 2$ бол нэг удаа солигдсонтой адил. Иймд зөвхөн чадвар бүрийг toggle хийх эсэхийг шийдэхэд хангалттай. □

Үүнээс гарах дүгнэлт:

Тодорхойлолт:
Бодлогыг $S \subseteq \{1, 2, \ldots, k\}$ дэд олонлогийг сонгох гэж хялбарчилж болно. Энд S нь toggle хийх чадваруудын олонлог. Нийт 2^k сонголт байна.

3 Нэг сонголтод тооцоолол

S дэд олонлогийг сонгосон гэе. i-р цэргийн хувьд:

- skills_i = i-р цэргийн чадваруудын олонлог
- $\mathrm{overlap}_i = \mathrm{skills}_i \cap S$ (давхцал)
- $|\mathrm{overlap}_i|$ тэгш бол цэрэг хэвээрээ, сондгой бол фронтоо солино

Цэрэг зүүн фронтод эцэслэх нөхцөл:

$$\mathrm{left}_i = \begin{cases} 1 \text{ хэрэв } (\mathrm{side}_i = L \text{ ба } |\mathrm{overlap}_i| \equiv 0) \text{ эсвэл } (\mathrm{side}_i = R \text{ ба } |\mathrm{overlap}_i| \equiv 1) \\ 0 \text{ бусад тохиолдолд} \end{cases} \pmod 2 \quad (1)$$

Нэг S-ийн хувьд бүх N цэргийг шалгахад $O(N)$ хугацаа шаардагдана (чадваруудын давхцлыг bitwise AND-ээр, popcount-ыг $O(1)$-д тооцно).

4 Brute force алгоритм

Бүх 2^k дэд олонлогийг туршиж, тус бүрд N цэргийг шалган, зүүн фронтод хамгийн олон цэрэг байх сонголтыг олно.

```
best = 0
for mask = 0 to 2^k - 1:
    left_count = 0
    for i = 0 to N-1:
        overlap = skills[i] AND mask
        flipped = popcount(overlap) mod 2
        if (side[i]=L and flipped=0) or (side[i]=R and flipped=1):
            left_count++
    best = max(best, left_count)
return best
```

5 Цагийн нарийвчлалын шинжилгээ

Теорем 1: Алгоритм нь $O(N)$ цагийн нарийвчлалтай
$k = 15$ нь бодлогын тогтмол (constant) учир алгоритмын цагийн нарийвчлал нь $O(N)$.

Баталгаа:
Алгоритмын нийт үйлдлийн тоог тооцъё.

Гадаад давталт: 2^k удаа давтагдана.
Дотоод давталт: Давталт бүрт N цэргийг шалгана.
Нэг цэргийн шалгалт: Bitwise AND, popcount, харьцуулалт — тус бүр $O(1)$.

Нийт үйлдлийн тоо:

$$T(N, k) = 2^k \cdot N \cdot O(1) = O\left(2^k \cdot N\right) \tag{2}$$

Энд $k = 15$ нь бодлогоор тогтмол (оролтоос хамаарахгүй). Тиймээс:

$$T(N) = O(2^{15} \cdot N) = O(32768 \cdot N) = O(N) \tag{3}$$

Big-O тэмдэглэгээнд тогтмол үржвэрийг хасдаг учир $O(c \cdot N) = O(N)$ (c тогтмол бол).

Ерөнхийдөө 2^k-г тогтмол гэж үзэж болох шалтгаан нь:
1. Бодлогын нөхцлөөр $k = 15$ нь үргэлж тогтмол
2. k нь оролтын хэмжээ N-ээс хамааралгүй
3. Тиймээс $2^k = 2^{15} = 32768$ нь тогтмол тоо

Энэ нь “fixed-parameter” нарийвчлалын ерөнхий зарчимд нийцнэ: хэрэв параметр k тогтмол бол $O(f(k) \cdot N)$ нь $O(N)$ болно. □

6 Анхааруулга: k тогтмол биш бол

Теорем 2: Ерөнхий тохиолдолд NP-хэцүү
Хэрэв k-г оролтын нэг хэсэг гэж үзвэл (өөрөөр хэлбэл k нь дурын утга авч болох бол), энэ бодлого нь MAX-XOR-SAT бодлогын хувилбар бөгөөд NP-хэцүү.

Баталгаа:
Цэрэг бүр $\boldsymbol{x} = (x_1, x_2, ..., x_k) \in \{0, 1\}^k$ шийдвэрийн хувьсагчдаар илэрхийлэгдэх XOR хязгаарлалт өгнө. Энд $x_j = 1$ бол j чадварыг toggle хийнэ гэсэн үг.

i-р цэргийн хувьд зүүн фронтод байх нөхцөл:

$$\text{side}_i \oplus \left(\bigoplus_{j \in \text{skills}_i} x_j \right) = 0 \quad (4)$$

Энд $\oplus$ нь XOR үйлдэл. Аль болох олон хязгаарлалтыг хангах $\boldsymbol{x}$ олох нь MAX-XOR-SAT бодлого бөгөөд NP-хэцүү гэдэг нь батлагдсан.

Гэхдээ манай бодлогод $k = 15$ тогтмол учир 2^k бүрэн хайлт нь бодитоор хийгдэх боломжтой. $\square$

7 Дүгнэлт

Арга	Цагийн нарийвчлал	Тайлбар
Brute force (k тогтмол)	$O(N)$	Манай шийдэл
Brute force (k хувьсагч)	$O(2^k \cdot N)$	Экспоненциал
Meet in the middle	$O(2^{k/2} \cdot N)$	Оновчлол

Энэхүү баталгаагаар brute force алгоритм нь $k = 15$ тогтмол нөхцөлд $O(N)$ шугаман цагийн нарийвчлалтай болохыг харууллаа. Тогтмол $2^{15} = 32768$ нь практикт хангалттай жижиг тоо бөгөөд $N < 10^3$ хязгаарлалттай хослоход нийт 32 сая үйлдэл хийгдэнэ.