Nim

Bergur Snorrason

23. mars 2023

- Skoðum einfaldan leik.
- Þú og einn andstæðingur eruð með hrúgu af n steinum.
- Þið skiptist á að taka einn eða tvo steina úr hrúgunni þar til hrúgan er tóm.
- Sá sem á að gera þegar hrúgan er tóm tapar.
- Hver vinnur?
  - Sá sem byrjar?
  - Sá sem byrjar ekki?
  - Fer eftir öðrum aðstæðum?

- Við samasömum stöður í leiknum við fjölda steina í hrúgunni.
- Við segjum því að 0 sé *tapstaða*.
- Staða er sögð *vinningsstaða* ef það er leikur sem breytir henni í tapstöðu.
- Almennt er staða er sögð tapstaða ef það er enginn leikur sem breytir henni í tapstöðu.
- Munið að við getum annaðhvort fjarlægt einn eða tvo steina, svo stöðurnar 1 og 2 eru vinningsstöður.
- Almennt fæst að staða n er tapstaða þá og því aðeins að n gangi upp í þrjá.

- Almennt fæst að staða n er tapstaða þá og því aðeins að n gangi upp í þrjá.
- Þetta fæst með þrepun.
- Gerum ráð fyrir að $3 \cdot k$ sé tapstaða.
- Þá eru stöður $3 \cdot k + 1$ og $3 \cdot k + 2$ vinningsstöður.
- Þetta eru líka einu stöðurnar sem maður kemst í frá stöðunni $3 \cdot (k + 1)$, svo hún er tapstaða.

- Tökum aðeins flóknara dæmi.
- Gerum ráð fyrir að við séum með m hrúgur með $n_1, \ldots, n_m$ steinum í hverri hrúgu.
- Nú skiptast keppendur á að velja hrúgu og taka eins marga steina úr henni og þeir vilja (að minnsta kosti einn).
- Aftur tapar sá sem getur ekki gert.
- Hvernig getum við lýst vinningsstöðunum og tapstöðunum?
- Þessi spurning er flóknari.
- Hér samsömum við m-dir við stöður.
- Hægt er að sýna að staðan $(k_1, \ldots, k_m)$ er tapstaða þá og því aðeins að $k_1 \oplus \cdots \oplus k_m = 0$.

- Gerum ráð fyrir að við séum með leik þar sem:
  - Engar huldar upplýsingar.
  - Leikurinn klárast eftir endanlegan fjölda leikja.
  - Fyrir tiltekna stöðu geta báðir spilarar gert það sama.
  - Leikurinn klárast þegar það er enginn löglegur leikur og sá sem á að gera tapar.
- Þá getum við úthlutað hverjum leik heiltölu endurkvæmt.
- Ef engin löglegur leikur er í boði fær staðan heiltöluna 0.
- Annars fær staðan minnstu jákvæðu heiltöluna sem er ekki hægt að komst í úr stöðunni.
- Við höfum þá að staða er tapstaða þá og því aðeins að henni sé úthlutað töluna 0.
- Þessar tölur kallast Grundy tölur staðanna.

- Gerum ráð fyrir að við séum að spila marga leiki í einu, þar sem hver leikur er eins og lýst var að ofan.
- Hver aðgerð felur þá í sér að velja einn af leikjunum sem er verið að spila og framkvæm leik í honum.
- Sá tapar sem lendir í því að eiga engan löglegan leik.
- Ef við gerum ráð fyrir að undirleikirnir hafi Grundy tölurnar $k_1, \ldots, k_m$ þá er Grundy tala sameinaða leiksins $k_1 \oplus \cdots \oplus k_m$.

- Tökum nú dæmi.
- Gerum ráð fyrir að við teiknum n kassa á blað í línu, nema það er búið að krota yfir kassann í miðjunni.
- Hver leikur felur í sér að krota yfir eins marga aðliggjandi kassa og manni lystir (að minnsta kosti einn).
- Sá sem getur ekki leikið tapar.

- Þegar við kortum yfir aðliggjandi reiti erum við í rauninni að skipta einum leik í tvo.
- Gerum ráð fyrir að það séu n reitir (númeraðir $1, 2, \ldots, n$) og að Grundy tala leiks fyrir k reiti sé g_k.
- Ef við krotum yfir reiti $i, i+1, \ldots, j-1, j$ skiptist leikurinn í tvennt, annar undirleikurinn með $i-1$ kassa og hinn með $n-j$ kassa.
- Grundy talan fyrir þennan nýja yfirleik er $g_{i-1} \oplus g_{n-j}$ ef $i > 1$, og g_{n-j} ef $i = 1$.

```c
int mex(int* a, int n)
{
    int b[n], i;
    for (i = 0; i < n; i++) b[i] = 0;
    for (i = 0; i < n; i++) if (a[i] < n) b[a[i]] = 1;
    for (i = 0; i < n; i++) if (!b[i]) break;
    return i;
}

int d[MAXN];
int dp_lookup(int x)
{
    if (x == 0) return 0;
    if (d[x] != -1) return d[x];
    int i, j, a[x*x + 1];
    for (a[0] = 0, i = 0; i < x; i++) for (j = i; j < x; j++)
        a[++a[0]] = dp_lookup(i)^dp_lookup(x - j - 1);
    return d[x] = mex(a + 1, a[0]);
}
```

```c
int d[MAXN];
int dp_lookup(int x)
{
    if (x == 0) return 0;
    if (d[x] != -1) return d[x];
    int i, j, a[x*x + 1];
    for (a[0] = 0, i = 0; i < x; i++) for (j = i; j < x; j++)
        a[++a[0]] = dp_lookup(i)^dp_lookup(x - j - 1);
    return d[x] = mex(a + 1, a[0]);
}

void finna_besta_leik(int *a, int n)
{
    int i, j, k, x, y;
    for (i = 0; i < n; i++) for (j = i; j < n; j++)
    {
        for (k = i; k <= j; k++) if (!a[k]) break;
        if (k <= j) continue;
        for (k = i; k <= j; k++) a[k] = 0;
        for (x = y = k = 0; k <= n; k++)
        {
            if (k == n || !a[k]) x ^= dp_lookup(y), y = 0;
            else y++;
        }
        if (!x) return;
        for (k = i; k <= j; k++) a[k] = 1;
    }
    for (k = 0; k < n; k++) if (a[k]) break;
    a[k] = 0;
}
```

- Til að finna Grundy tölu á stöðu í einfalda leiknum þurfum við að reikna n gildi, og hvert gildi er reiknað í $\mathcal{O}(n^2)$ tíma.
- Svo það tekur $\mathcal{O}(n^3)$ tíma að reikna allar Grundy tölurnar.
- Til að finna best leikinn í hverri stöðu prófum við alla mögulega leiki, sem tekur $\mathcal{O}(n^3)$ tíma.
- Í heildina tekur þetta því $\mathcal{O}(n^3)$ tíma.