Lausnir á dæmum tengd viku fjögur

Bergur Snorrason

9. febrúar 2022

- Ég mun leysa eftirfarandi dæmi:

- Ég mun leysa eftirfarandi dæmi:
  - *Kaffi*,

- Ég mun leysa eftirfarandi dæmi:
  - *Kaffi*,
  - *Bad Packing*.

Kaffi

- Við erum með litla kassa sem við viljum stafla við vegg sem er með takmarkaða breidd.

Kaffi

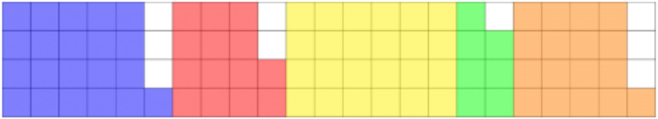
- Við erum með litla kassa sem við viljum stafla við vegg sem er með takmarkaða breidd.
- Hver kassi hefur lit og kassar af mismunandi lit mega ekki staflast ofan á hvorn annan.

Kaffi

- Við erum með litla kassa sem við viljum stafla við vegg sem er með takmarkaða breidd.
- Hver kassi hefur lit og kassar af mismunandi lit mega ekki staflast ofan á hvorn annan.
- Við viljum lágmarka svæðið á veggnum sem sést fyrir aftan stöfluðu kassana.

Kaffi

- Við erum með litla kassa sem við viljum stafla við vegg sem er með takmarkaða breidd.
- Hver kassi hefur lit og kassar af mismunandi lit mega ekki staflast ofan á hvorn annan.
- Við viljum lágmarka svæðið á veggnum sem sést fyrir aftan stöfluðu kassana.



Kaffi

- Látum gefnu breiddina vera w.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.
- En kassarnir hafa fast flatarmál, svo okkur nægir að lágmarka heildarflatarmálið.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.
- En kassarnir hafa fast flatarmál, svo okkur nægir að lágmarka heildarflatarmálið.
- Þar sem breddin er föst nægir okkur að lágmarka hæðina.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.
- En kassarnir hafa fast flatarmál, svo okkur nægir að lágmarka heildarflatarmálið.
- Þar sem breddin er föst nægir okkur að lágmarka hæðina.
- Svo okkur nægir að finna minnstu hæðina h þannig að það megi stafla kössunum í w stafla og hver stafli er hefur mest h kassa.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.
- En kassarnir hafa fast flatarmál, svo okkur nægir að lágmarka heildarflatarmálið.
- Þar sem breddin er föst nægir okkur að lágmarka hæðina.
- Svo okkur nægir að finna minnstu hæðina h þannig að það megi stafla kössunum í w stafla og hver stafli er hefur mest h kassa.
- Tökum þó eftir að ef þetta er hægt fyrir h þá er þetta líka hægt fyrir $h' > h$.

Kaffi

- Látum gefnu breiddina vera w.
- Flatarmálið sem við viljum lágmarka er heildarflatarmálið mínus flatarmál kassana.
- En kassarnir hafa fast flatarmál, svo okkur nægir að lágmarka heildarflatarmálið.
- Þar sem breddin er föst nægir okkur að lágmarka hæðina.
- Svo okkur nægir að finna minnstu hæðina h þannig að það megi stafla kössunum í w stafla og hver stafli er hefur mest h kassa.
- Tökum þó eftir að ef þetta er hægt fyrir h þá er þetta líka hægt fyrir $h' > h$.
- Svo við getum notað helmingunarleit til að finn minnstu hæðina.

Kaffi

- Hvernig athugum við hvort gefin hæð h sé nógu há.

Kaffi

- Hvernig athugum við hvort gefin hæð h sé nógu há.
- Gerum ráð fyrir kassarnir hafi n mismunandi liti og a_i kassar séu af i-ta litnum.

Kaffi

- Hvernig athugum við hvort gefin hæð h sé nógu há.
- Gerum ráð fyrir kassarnir hafi n mismunandi liti og a_i kassar séu af i-ta litnum.
- Við þurfum þá $\lfloor a_i/h \rfloor$ stafla fyrir kassa af lit i.

Kaffi

- Hvernig athugum við hvort gefin hæð h sé nógu há.
- Gerum ráð fyrir kassarnir hafi n mismunandi liti og a_i kassar séu af i-ta litnum.
- Við þurfum þá $\lfloor a_i/h \rfloor$ stafla fyrir kassa af lit i.
- Svo h er nógu stórt ef

$$\sum_{i=1}^{n} \lfloor a_i/h \rfloor \leq w.$$

Kaffi

```c
#include <stdio.h>
#define MAXN 100000
typedef long long ll;

int main()
{
    ll i, n, w, r, s, t = 0, a[MAXN];
    scanf("%lld%lld", &n, &w);
    for (i = 0; i < n; i++) scanf("%lld", &a[i]);
    for (i = 0; i < n; i++) t += a[i];
    r = 1, s = 1000000001;
    while (r < s)
    {
        ll m = (r + s)/2, z = 0;
        for (i = 0; i < n; i++) z += (a[i] + m - 1)/m;
        if (z > w) r = m + 1;
        else s = m;
    }
    printf("%lld\n", w*r - t);
    return 0;
}
```

Kaffi

- Við erum $\mathcal{O}(\quad)$ tíma að ganga úr skugga um það hvort tiltekin hæð sé nógu há.

Kaffi

- Við erum $\mathcal{O}(n)$ tíma að ganga úr skugga um það hvort tiltekin hæð sé nógu há.

Kaffi

- Við erum $\mathcal{O}(\, n \,)$ tíma að ganga úr skugga um það hvort tiltekin hæð sé nógu há.
- Með helmingunarleitinn verður tímaflækjan $\mathcal{O}(\qquad\qquad)$, þar sem m er mesti mögulegi fjöldi stóla af hverjum lit.

Kaffi

- Við erum $\mathcal{O}(\ n\ )$ tíma að ganga úr skugga um það hvort tiltekin hæð sé nógu há.
- Með helmingunarleitinn verður tímaflækjan $\mathcal{O}(n \log m)$, þar sem m er mesti mögulegi fjöldi stóla af hverjum lit.

Bad Packing

- Við erum með $n \leq 10^3$ hluti, hlutirnir hafa þyngdir $w_1, \ldots, d_n$ og bakpoka sem rúmar $c \leq 10^5$ samtals þyngd.

Bad Packing

- Við erum með $n \leq 10^3$ hluti, hlutirnir hafa þyngdir $w_1, \ldots, d_n$ og bakpoka sem rúmar $c \leq 10^5$ samtals þyngd.
- Ef við veljum hluti af handahófi þangað til bakpokinn okkar rúmar ekki fleiri hluti, hver er þá minnsta þyngdin sem við getum endað með.

Bad Packing

- Við erum með $n \leq 10^3$ hluti, hlutirnir hafa þyngdir $w_1, \ldots, d_n$ og bakpoka sem rúmar $c \leq 10^5$ samtals þyngd.
- Ef við veljum hluti af handahófi þangað til bakpokinn okkar rúmar ekki fleiri hluti, hver er þá minnsta þyngdin sem við getum endað með.
- Til dæmis, ef $n = 3$, $c = 6$ og við hofum þyngdir 3, 3 og 4 þá gætum við endað með þyngdir $3 + 3 = 6$ og 5.

Bad Packing

- Þetta dæmi minnir mjög á hlutmengjasummu dæmið.

Bad Packing

- Þetta dæmi minnir mjög á hlutmengjasummu dæmið.
- Rifjum upp að við létum $f(i,j)$ vera 1 ef einhver hlutruna $w_1, \ldots, w_i$ hafi summu j og f hefur rakningarformúlu

$$f(i,j) = \begin{cases} 1, & \text{ef } i = 0 \text{ og } j = 0, \\ 0, & \text{ef } i = 0 \text{ og } j \neq 0, \text{ eða } j < 0, \\ f(i-1,j) \text{ eða} & \\ \quad f(i-1,j-a_i)), & \text{ef } i \neq 0. \end{cases}$$

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.
- Ef við tökum ekki þá hluti þá gætum við bætt þeim í bakpokann okkar.

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.
- Ef við tökum ekki þá hluti þá gætum við bætt þeim í bakpokann okkar.
- Látum s vera summu allra hlutanna sem hafa þyngd minni eða jafna k.

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.
- Ef við tökum ekki þá hluti þá gætum við bætt þeim í bakpokann okkar.
- Látum s vera summu allra hlutanna sem hafa þyngd minni eða jafna k.
- Við þurfum þá að svara spurningunni: „Getum við valið hluti, alla þyngri en k, sem hafa summu $k - s$?".

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.
- Ef við tökum ekki þá hluti þá gætum við bætt þeim í bakpokann okkar.
- Látum s vera summu allra hlutanna sem hafa þyngd minni eða jafna k.
- Við þurfum þá að svara spurningunni: „Getum við valið hluti, alla þyngri en k, sem hafa summu $k - s$?".
- Þetta er hlutmengjasummudæmi, en við þurfum þó að passa okkur.

Bad Packing

- Reynum að svara spurningunni: „Er mögulegt að enda með þyngd k?".
- Tökum eftir að við þurfum alltaf að taka alla hluti sem hafa þyngd minni eða jafna k.
- Ef við tökum ekki þá hluti þá gætum við bætt þeim í bakpokann okkar.
- Látum s vera summu allra hlutanna sem hafa þyngd minni eða jafna k.
- Við þurfum þá að svara spurningunni: „Getum við valið hluti, alla þyngri en k, sem hafa summu $k - s$?".
- Þetta er hlutmengjasummudæmi, en við þurfum þó að passa okkur.
- Ef við notum beint kóðann úr síðasta fyrirlestri og gerum þetta fyrir hverja þyngd fæst tímaflækjan $\mathcal{O}(nc^2)$, sem er alltof hægt.

Bad Packing

- Þessi slæma tímaflækja fæst þó því við leysum hlutmengjasummudæmið c sinnum.

Bad Packing

- Þessi slæma tímaflækja fæst þó því við leysum hlutmengjasummudæmið c sinnum.
- Rifjum upp að

$$f(i, j) = \begin{cases} 1, & \text{ef } i = 0 \text{ og } j = 0, \\ 0, & \text{ef } i = 0 \text{ og } j \neq 0, \text{ eða } j < 0, \\ f(i-1, j) \text{ eða} & \\ \quad f(i-1, j-a_i)), & \text{ef } i \neq 0. \end{cases}$$

Bad Packing

- Þessi slæma tímaflækja fæst þó því við leysum hlutmengjasummudæmið c sinnum.
- Rifjum upp að

$$f(i,j) = \begin{cases} 1, & \text{ef } i = 0 \text{ og } j = 0, \\ 0, & \text{ef } i = 0 \text{ og } j \neq 0, \text{ eða } j < 0, \\ f(i-1,j) \text{ eða} & \\ \quad f(i-1, j-a_i)), & \text{ef } i \neq 0. \end{cases}$$

- Tökum eftir að eftir að hafa leyst hlutmengjasummu dæmið einu sinni höfum við í raun leyst það fyrir öll söfn vigta af gerðinni $w_1, \ldots, w_k$.

Bad Packing

- Þessi slæma tímaflækja fæst þó því við leysum hlutmengjasummudæmið c sinnum.
- Rifjum upp að

$$f(i,j) = \begin{cases} 1, & \text{ef } i = 0 \text{ og } j = 0, \\ 0, & \text{ef } i = 0 \text{ og } j \neq 0, \text{ eða } j < 0, \\ f(i-1,j) \text{ eða} & \\ \quad f(i-1, j-a_i)), & \text{ef } i \neq 0. \end{cases}$$

- Tökum eftir að eftir að hafa leyst hlutmengjasummu dæmið einu sinni höfum við í raun leyst það fyrir öll söfn vigta af gerðinni $w_1, \ldots, w_k$.
- Ef við röðum vigtunum okkar í minnkandi röð þurfum við bara að leysa hlutmengjasummudæmið einu sinni.

Bad Packing

```c
int d[MAXN][MAXC], a[MAXN];
int foo(int x, int y)
{
    if (x < 0 || y < 0) return !y;
    if (d[x][y] != -1) return d[x][y];
    return d[x][y] = foo(x - 1, y) || foo(x - 1, y - a[x]);
}

int main()
{
    int i, j, t, n, c;
    scanf("%d%d", &n, &c);
    for (i = 0; i < n; i++) scanf("%d", &a[i]);
    for (i = 0; i < n; i++) for (j = 0; j < c + 1; j++) d[i][j] = -1;
    qsort(a, n, sizeof *a, cmp);
    for (i = 0; i < c + 1; i++)
    {
        for (t = 0, j = n - 1; j >= 0; j--)
        {
            if (a[j] > c - i) break;
            t += a[j];
        }
        if (t <= i && foo(j, i - t)) break;
    }
    printf("%d\n", i);
    return 0;
}
```

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(\quad)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(\quad)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(c \cdot n)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(c \cdot n)$.
- Við höfum að lokum tvöfalda `for`-lykkju, sú ytri af lengd $\mathcal{O}(\ c\ )$ og sú innri af lengd $\mathcal{O}(\ n\ )$, og samtals er þetta $\mathcal{O}(\qquad)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(c \cdot n)$.
- Við höfum að lokum tvöfalda `for`-lykkju, sú ytri af lengd $\mathcal{O}(\ c\ )$ og sú innri af lengd $\mathcal{O}(\ n\ )$, og samtals er þetta $\mathcal{O}(c \cdot n)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(c \cdot n)$.
- Við höfum að lokum tvöfalda `for`-lykkju, sú ytri af lengd $\mathcal{O}(\ c\ )$ og sú innri af lengd $\mathcal{O}(\ n\ )$, og samtals er þetta $\mathcal{O}(c \cdot n)$.
- Samtals er þetta forrit því $\mathcal{O}(\qquad)$.

Bad Packing

- Við röðum n tölum sem tekur $\mathcal{O}(n \log n)$.
- Við þurfum líka að leysa hlutmengjasummudæmið, sem tekur $\mathcal{O}(c \cdot n)$.
- Við höfum að lokum tvöfalda `for`-lykkju, sú ytri af lengd $\mathcal{O}(\ c\ )$ og sú innri af lengd $\mathcal{O}(\ n\ )$, og samtals er þetta $\mathcal{O}(c \cdot n)$.
- Samtals er þetta forrit því $\mathcal{O}(c \cdot n)$.