

# Miðmisseriskönnun

Bergur Snorrason

22. febrúar 2021

- ▶ Miðmisseriskönnun var að ljúka.

- ▶ Miðmisseriskönnun var að ljúka.
- ▶ Það er ein athugasemd sem ég vil ræða.

Mætti kenna aðeins meira markvisst, taka fyrir alla algóritmana sem maður þarf að kunna fyrir næstu vikudæmi. Oft eru einhverjar pælingar sem maður hefur aldrei séð áður, eins og svokallaða „Two egg” vandamálið sem var í raun endurskrifað í dæmið „Exploding Batteries” á Kattis. Þá þurfti maður að eyða miklum tíma í að googla og finna svipað vandamál (þannig rakst ég á „Two egg problem”).

```

1 #include <stdio.h>
2 #define rep(E, F) for (E = 0; E < (F); E++)
3 int min(int a, int b) { if (a > b) return b; return a; }
4 int max(int a, int b) { if (a < b) return b; return a; }
5
6 int d[5000], n;
7 int foo(int x)
8 {
9     if (x == 1) return 0;
10    if (d[x] != -1) return d[x];
11    int i;
12    d[x] = (1 << 30);
13    rep(i, x) if (i != 0) d[x] = min(d[x], max(i, 1 + foo(x - i)));
14    return d[x];
15 }
16
17 int main()
18 {
19     int i;
20     rep(i, 5000) d[i] = -1;
21     scanf("%d", &n);
22     while (n != 0)
23     {
24         printf("%d\n", foo(n));
25         scanf("%d", &n);
26     }
27     return 0;
28 }

```

- ▶ Ég mun fara að nota styttingunar `rep(...)` restina af námskeiðinu.

- ▶ Ég mun fara að nota styttingunar `rep(...)` restina af námskeiðinu.
- ▶ Hún er notuð til að spara skriftir í for-lykkjum.

- ▶ Ég mun fara að nota styttingunar `rep(...)` restina af námskeiðinu.
- ▶ Hún er notuð til að spara skriftir í for-lykkjum.
- ▶ Hún er skilgreind með `#define rep(E, F) for (E = 0; E < (F); E++)`.



- ▶ Ég mun fara að nota styttingunar `rep(...)` restina af námskeiðinu.
- ▶ Hún er notuð til að spara skriftir í for-lykkjum.
- ▶ Hún er skilgreind með `#define rep(E, F) for (E = 0; E < (F); E++)`.
- ▶ Þetta þýðir að `rep(i, n)` er jafngilt því að skrifa for (`i = 0; i < n; i++`).

- ▶ Ég mun fara að nota styttingunar `rep(...)` restina af námskeiðinu.
- ▶ Hún er notuð til að spara skriftir í for-lykkjum.
- ▶ Hún er skilgreind með `#define rep(E, F) for (E = 0; E < (F); E++)`.
- ▶ Þetta þýðir að `rep(i, n)` er jafngilt því að skrifa for (`i = 0; i < n; i++`).
- ▶ Sem dæmi um hagnýtingu þessara styttingu má sjá útfærsluna mína á bakstrengsfylkjasmíð (e. suffix array constructor).

```

1 void suffix_array(char* s, int n, int* p)
2 {
3     s[n] = ' ' - 1, s[n + 1] = '\0'; n++;
4     int i, k = 1, a = 256, h;
5     int c[n], g[a], gg[n], pn[n], cn[n];
6     rep(i, a) g[i] = 0;
7     rep(i, n) g[s[i]]++;
8     rep(i, a) if (i != 0) g[i] += g[i - 1];
9     rep(i, n) p[--g[s[i]]] = i;
10    c[p[0]] = 0;
11    rep(i, n) if (i != 0) c[p[i]] = (s[p[i]] != s[p[i - 1]] ? ++k : k) - 1;
12    for (h = 0; (1 << h) < n; ++h)
13    {
14        rep(i, n) pn[i] = p[i] - (1 << h), pn[i] += pn[i] < 0 ? n : 0;
15        rep(i, k) gg[i] = 0;
16        rep(i, n) gg[c[pn[i]]]++;
17        rep(i, k) if (i != 0) gg[i] += gg[i - 1];
18        rep(i, n) p[--gg[c[pn[n - 1 - i]]]] = pn[n - 1 - i];
19        cn[p[0]] = 0;
20        k = 1;
21        rep(i, n) if (i != 0) cn[p[i]] =
22            (c[p[i]] != c[p[i - 1]] ||
23             c[(p[i] + (1 << h)) % n] !=
24              c[(p[i - 1] + (1 << h)) % n] ? ++k : k) - 1;
25        rep(i, n) c[i] = cn[i];
26    }
27    rep(i, n - 1) p[i] = p[i + 1];
28 }

```

