

- (c) Ha minden m bemenet esetén A legalább m lépést tesz, akkor A biztosan nem polinomiális algoritmus.
- (d) Ha van olyan m , amelyre A legalább m lépést tesz, akkor A biztosan nem polinomiális algoritmus.
- (e) Ha minden páros m bemenet esetén A legalább m lépést tesz, akkor A biztosan nem polinomiális algoritmus.

11. Az alábbi C kódok közül az első $\lfloor \sqrt{n} \rfloor$ -t, a második $\lfloor \log_2 n \rfloor$ -t számítja ki bármely bemenetként (10-es számrendszerben) kapott $n > 0$ egész esetén. Tegyük fel, hogy a kódok végrehajtásakor a gép az (alsó tagozatban tanult) „írásbeli” összeadás, szorzás, stb. segítségével végzi el. Döntsük el, hogy az eljárások polinomiálisak-e.

```
(a)  x = 0; y = 0;
      while (y <= n) {
          x = x+1;
          y = x*x;
      }
      printf("Eredmény: %d", x-1);
```

```
(b)  x = 0; y = 1;
      while (y <= n) {
          x = x+1;
          y = 2*y;
      }
      printf("Eredmény: %d", x-1);
```

12. Az alábbi C kód a bemenetként (10-es számrendszerben) kapott $0 < a < n$ egészek esetén az n -nek az a -nál nemnagyobb osztói közül a legnagyobbat számítja ki. Tegyük fel, hogy a kód végrehajtásakor a gép az (alsó tagozatban tanult) „írásbeli” összeadás, szorzás, stb. segítségével végzi el. Döntsük el, hogy az eljárás polinomiális-e.

```
while (n%a != 0) {
    a = a-1;
}
printf("Eredmény: %d", a);
```