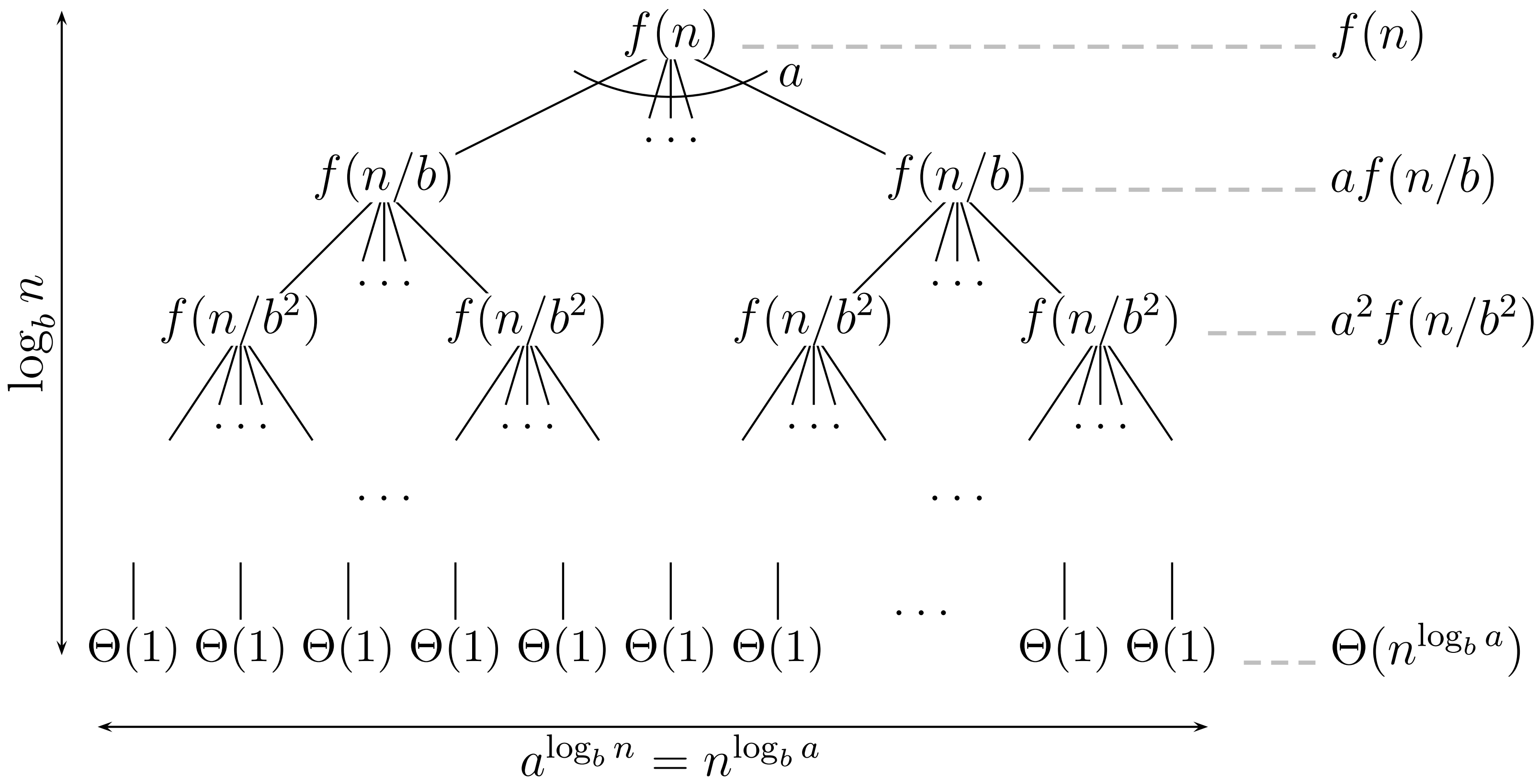
Master theorem:

Nech $T(n) = aT(n/b) + f(n)$, $T(1) = \Theta(1)$. Nech $k = \log_b a$. Potom:

1. Ak $\boxed{f(n) \in O(n^{k-\varepsilon})}$ pre niektoré $\varepsilon > 0$, potom $\boxed{T(n) \in \Theta(n^k)}$.
2. Ak $\boxed{f(n) \in \Theta(n^k)}$, potom $\boxed{T(n) \in \Theta(f(n) \log n)}$.
3. Ak $\boxed{f(n) \in \Omega(n^{k+\varepsilon})}$ pre niektoré $\varepsilon > 0$ a platí podmienka regularity, potom $\boxed{T(n) \in \Theta(f(n))}$.

Podmienka regularity: Existuje $c < 1$ také, že pre všetky dostatočne veľké n platí $af(n/b) \leq cf(n)$.



$f(n)$
a
$f(n/b)$
$f(n/b)$
$f(n/b^2)$
$f(n/b^2)$
$f(n/b^2)$
$f(n/b^2)$
$\Theta(1)$
$f(n)$
$af(n/b)$
$a^2f(n/b^2)$
$\Theta(n^{\log_b a})$
$\log_b n$
$a^{\log_b n} = n^{\log_b a}$

$$T(n) = \Theta(n^k) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

Prípad 1: $f(n) \in O(n^{k-\varepsilon})$ **= "ťažký spodok"**

$$\sum_{j=0}^{\log_b n - 1} a^j f(n/b^j) =$$

Teda $\sum_{j=0}^{\log_b n - 1} a^j f(n/b^j) = O(n^k) \Rightarrow T(n) = \Theta(n^k)$

$$T(n) = \Theta(n^k) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

Prípad 2: $f(n) \in \Theta(n^k)$ **= "vyvážené úrovne"**

Chceme ukázať, že každá úroveň "prispeje" prácou $\Theta(n^k)$

$\Rightarrow T(n) = \Theta(n^k \log n)$

$$T(n) = \Theta(n^k) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

Prípad 3: $f(n) \in \Omega(n^{k+\varepsilon})$ **= "ťažký vrch"**

Podmienka regularity: $af(n/b) \leq cf(n) \Rightarrow a^j f(n/b^j) \leq c^j f(n)$

$\Rightarrow T(n) = \Theta(f(n))$

Master theorem:

Nech $T(n) = aT(n/b) + f(n)$, $T(1) = \Theta(1)$. Nech $k = \log_b a$. Potom:

1. Ak $\boxed{f(n) \in O(n^{k-\varepsilon})}$ pre niektoré $\varepsilon > 0$, potom $\boxed{T(n) \in \Theta(n^k)}$.
2. Ak $\boxed{f(n) \in \Theta(n^k)}$, potom $\boxed{T(n) \in \Theta(f(n) \log n)}$.
3. Ak $\boxed{f(n) \in \Omega(n^{k+\varepsilon})}$ pre niektoré $\varepsilon > 0$ a platí podmienka regularity, potom $\boxed{T(n) \in \Theta(f(n))}$.

Podmienka regularity: Existuje $c < 1$ také, že pre všetky dostatočne veľké n platí $af(n/b) \leq cf(n)$.

Najbližší pár bodov

Úloha: Daných je n bodov $p_1, \ldots, p_n$ $(p_i = (x_i, y_i))$
Nájdite pár bodov, ktoré sú k sebe najbližšie (Euklidovská vzdialenosť)

Triviálne riešenie: Porovnaj $d(p_i, p_j)$ pre každý pár bodov
$\Rightarrow$ čas $\Theta(n^2)$

Lepší nápad: rozdeľuj a panuj

- Rozdeľ vertikálnou čiarou na dve rovnaké polovice P_L a P_R
- Rekurzívne nájdi:
  δ_L : najbližší pár v P_L
  δ_R : najbližší pár v P_R
- Spočítaj δ_M : najbližší pár "cez čiaru"
- Výsledok je $\min\{\delta_L, \delta_R, \delta_M\}$

$T(n) = 2T(n/2) + \Theta(n^2) = \Theta(n^2)$ (master theorem)

Šikovnejšie hľadanie bodov “cez čiaru”

Ak $\delta = \min\{\delta_L, \delta_R\}$, tak nás nezaujímajú body, ktoré sú od hranice ďalej ako δ!

```
function closest_pair(l,r)
  // Find the closest pair in P[l..r]
  // assume P[l..r] is sorted by x-coordinate
  if size(P)<2 then return infinity;
  // Divide: midx will be a dividing line
  mid:=(l+r)/2; midx:=P[mid].x;
  dl:=closest_pair(l,mid); dr:=closest_pair(mid+1,r);
  // as a side effect, P[l..mid] and P[mid+1..r]
  // are now sorted by y-coordinate
  delta:=min(dl,dr);
  QL:=select_candidates(l,mid,delta,midx);
  QR:=select_candidates(mid+1,r,delta,midx);
  dm:=delta_m(QL,QR,delta);
  // use merge make P[l..r] sorted by y-coordinate
  merge(l,mid,r);
  return min(dm,dl,dr);
```

```
function select_candidates(l,r,delta,midx)
  // From P[l..r] select all points which are
  // in the distance at most delta from midx line
  create empty array Q;
  for i:=l to r do
    if (abs(P[i].x-midx)<=delta)
      add P[i] to Q;
  return Q;
```

```
function delta_m(QL,QR,delta)
  // Are there two points p in QL, q in QR such that
  // d(p,q)<=delta? Return closest such pair.
  // Assume QL and QR are sorted by y coordinate
  j:=1; dm:=delta;
  for i:=1 to size(QL) do
    p:=QL[i];
    // find the bottom-most candidate from QR
    while (j<=n and QR[j].y<p.y-delta) do
      j:=j+1;
    // check all candidates from QR starting with j
    k:=j;
    while (k<=n and QR[k].y<=p.y+delta) do
      dm:=min(dm,d(p,QR[k]));
      k:=k+1;
  return dm;
```

```
//-------- main ---------
// P contains all the points
sort P by x-coordinate;
return closest_pair(1,n);
```

Časová zložitosť: Nech $T(n)$ je čas potrebný na vyriešenie problému pre n bodov.

- Rozdeľuj: $\Theta(1)$
- Panuj: $2T(n/2)$
- Skombinuj: $\Theta(n)$

Teda $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$ (master theorem).