



## هرم

### تعریف

هرم بیشینه (یا کمینه) یک داده‌ساختار درختی است با این ویژگی که هر مقدار متناظر هر رأس بیشتر (کمتر) یا مساوی فرزندهای خود است. بدین ترتیب عنصر بیشینه (کمینه) در ریشه قرار می‌گیرد. هرم‌های مختلف عملیات‌های مختلفی را پشتیبانی می‌کنند. عملیات‌های پایه‌ای یک هرم بیشینه عبارتند از: یافتن عنصر بیشینه، حذف عنصر بیشینه و درج

### پیاده‌سازی

هرم معمولاً با آرایه پیاده‌سازی می‌شود و نیاز به اشاره‌گر ندارد. اگر ریشه‌ی هرم را اندیس ۱ در نظر بگیریم، می‌توانیم فرزندهای اندیس  $x$  را  $2x$  و  $2x + 1$  فرض کنیم.

عنصر بیشینه در ریشه قرار دارد. برای درج، عنصر جدید را در آخر هرم افزوده و تا جایی که باید آن را بالا می‌آوریم. برای حذف عنصر بیشینه، آن را با آخرین عنصر هرم جابه‌جا می‌کنیم و سپس سعی می‌کنیم شرط هرم را برقرار کنیم.

```
int find_max() {
    return a[1];
}

void insert(int val) {
    a[++n] = val;
    for (int x = n; a[x / 2] < a[x]; x /= 2)
        swap(a[x], a[x / 2]);
}

int big_child(int x) {
    if (2 * x + 1 <= n && a[2 * x + 1] > a[2 * x])
        return 2 * x + 1;
    else if (2 * x <= n)
        return 2 * x;
    else
        return -1;
}

void delete_max() {
    swap(a[1], a[n--]);
    for (int x = 1; big_child(x) != -1 && a[x] < a[big_child(x)]; x = big_child(x))
        swap(a[x], a[big_child(x)]);
}
```

### کاربردها

- برای پیاده‌سازی صف اولویت معمولاً از هرم استفاده می‌شود.
- بعضی از انواع پیاده‌سازی الگوریتم دایکسترا یا درخت فراگیر کمینه از هرم استفاده می‌کنند.
- با درج تمام عناصر یک مجموعه در هرم کمینه و استخراج ریشه به طور مکرر، می‌توان اعضای مجموعه را مرتب کرد. به این الگوریتم، مرتب‌سازی هرمی می‌گویند.