Chương 8: Kỹ thuật thỏa hiệp không-thời gian

Sắp xếp đếm (Countingsort)

Giải thuật

```
CountingSort(a[0 .. n - 1]) {
   count[0 .. n - 1] = 0;
   for (i = 0; i < n - 1; i++)
      for (j = i + 1; j < n; j++)
         if (a[i] < a[j])
            count[j]++;
         else
            count[i]++;
   for (i = 0; i < n - 1; i++)
      b[count[i]] = a[i];
   a[0 .. n - 1]  = b[0 .. n - 1];
}
```

Distribution counting

Giải thuật

```
DistributionCounting(a[0 .. n - 1]) {
   f[0 .. high] = 0;
   for (i = 0; i < n; i++)
      f[a[i]] ++;
   for (i = 1; i <= high; i++)
      f[i] += f[i - 1];

   for (i = n - 1; i >= 0; i--) {
      b[f[a[i]] - 1] = a[i];
      f[a[i]]--;
   }
   a[0 .. n - 1]  = b[0 .. n - 1];
}
```

Giải thuật

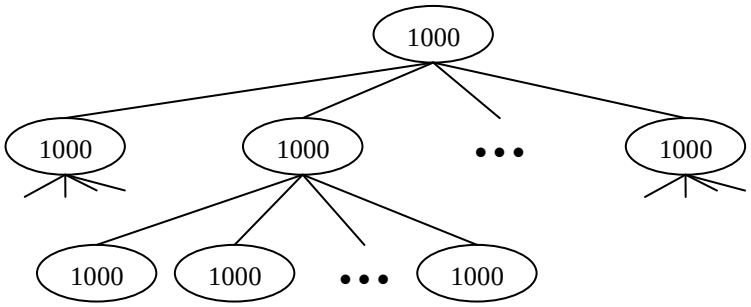
```
DistributionCountingSort(a[0 .. n - 1]) {
   f[0 .. hi - lo] = 0;
   for (i = 0; i < n; i++)
      f[a[i] - lo] ++;
   for (i = 1; i <= hi - lo; i++)
      f[i] += f[i - 1];
   for (i = n - 1; i >= 0; i--) {
      j = a[i] - lo;
      b[f[j] - 1] = a[i];
      f[j]--;
   }
   a[0 .. n - 1]  = b[0 .. n - 1];
}
```

Tìm kiếm chuỗi sử dụng DFA

Giải thuật

```
void     DFA_Search(char T[], int n, char P[], int m) {
   int   dfa[|P|+1][|Σ|] = {0};
   ComputeTransitionTable(P, m, Σ, dfa);
   q = 0;
   for (i = 0; i < n; i++) {
      q = dfa[q][T[i]];
      if (q == m)
         printf("Mẫu xuất hiện tại vị trí %d", i – m + 1);
   }
}
void  ComputeTransitionTable(P, m, Σ, dfa[|P|+1][|Σ|]) {
   for (q = 0; q <= m; q++)
      for (với mỗi a ∈ Σ) {
         k = min(m + 1, q + 2);
         repeat
            k--;
         until Pk ▽ Pqa;
         dfa[q][a] = k;
      }
}
```

Truy xuất với B-cây



1 nút
(1000 khóa)

1 × 1001 nút
(1000 × 1001 =
1,001,000 khóa)

1001 × 1001 =1002001 nút
(1000 × 1002001 =
1,002,001,000 khóa)

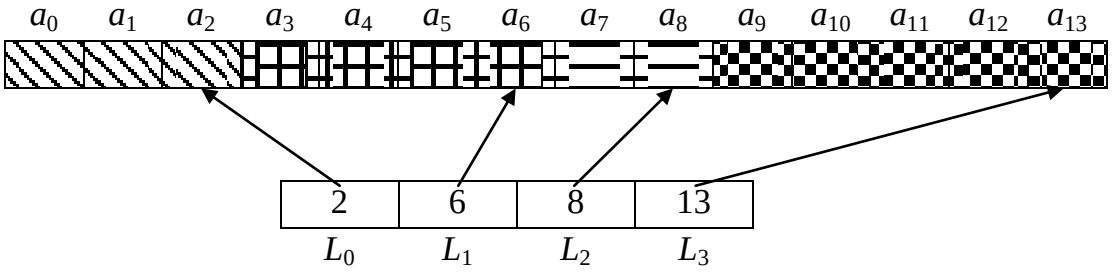
Sắp xếp "chớp nhoáng" (Flashsort)

Giai đoạn 1: Phân loại các phần tử

$$k_{a_i} = \left\lfloor \frac{(m-1)(a_i - min_a)}{max_a - min_a} \right\rfloor$$

```
for (i = 0; i < n; i++) {
   k = (m - 1) * (1.0 * (a[i] - min) / (max - min));
   L[k] += 1;
}
L[1]--;  // Chi so cua mang tinh tu 0
for (k = 1; k < m; k++)
   L[k] += L[k - 1];
```

Ví dụ: Mảng có 14 phần tử, chia làm $m = 4$ phân lớp



Giai đoạn 2: Phân lớp các phần tử

Giải thuật

```
count = 0;
i     = 0;
k     = m - 1;
while (count < n) {
   while (i > L[k]) {
      i++;
      k = (m - 1) * (1.0 * (a[i] - min) / (max - min));
   }
   flash = a[i];
   while (i <= L[k]) {
      k = (m - 1) * (1.0 * (flash - min) / (max - min));

      t     = a[L[k]];
      a[L[k]] = flash;
      flash = t;

      L[k]--;
      count++;
   }
}
```

Giai đoạn 3: Sắp xếp các phần tử trên từng phân lớp

Giải thuật

```
for (k = 1; k < m; k++)
   for (i = L[k] - 1; i > L[k - 1]; i--)
      if (a[i] > a[i + 1]) {
         t = a[i];
         j = i;
         while (t > a[j + 1])
            a[j++] = a[j + 1];
         a[j] = t;
      }
```