Exhaustive Search and Backtracking

Hao Jiang
Computer Science Department
Exhaustive Search

- Modern computers can compute millions of times per second.
- Exhaustive search is possible for some small scale problems.
- We study some systematic way to write such programs.
An Example

Combination:

Given a set \( \{1, 2, 3, 4, 5\} \), how many subsets does it have?

\[
\begin{align*}
{} \\
\{1\}, \{2\}, \{3\}, \ldots \\
\{1,2\}, \{1,3\}, \ldots \\
\{1,2,3\}, \ldots \\
\ldots \\
\{1,2,3,4,5\}
\end{align*}
\]
The Java Code

```java
public class MyCombine {
    public static void all_sub_string(int[] a, int n) {
        if (n == a.length) {
            for (int i = 0; i < a.length; i++)
                System.out.print( a[i] + " ");
            System.out.println();
            return;
        }

        a[n] = 0;
        all_sub_string(a, n+1);
        a[n] = 1;
        all_sub_string(a, n+1);
    }

    public static void main(String[] s) {
        int[] a = new int[3];
        all_sub_string(a, 0);
    }
}
```