Examples of Recursive Functions

GCD

- Using Euclid's method ($m \geq n > 0$):

$$\text{GCD}(m, n) = \begin{cases} n, \text{ if } m\%n = 0 \\ \text{GCD}(n, m\%n), \text{ otherwise} \end{cases}$$

- Dijkstra's method (assuming $m > n > 0$) GCD(m, n) is same as GCD$(m - n, n)$:

$$\text{GCD}(m, n) = \begin{cases} n, \text{ if } m = n \\ \text{GCD}(m - n, n), \text{ if } m > n \\ \text{GCD}(m, n - m), \text{ if } n > m \end{cases}$$

Examples of Recursive Functions

GCD: Euclid's Method

```
#include <stdio.h>
int GCD(int m, int n) {
    if ((m % n) == 0)
        return n;
    return GCD(n, m % n);
}

int main() {
    int m, n;

    printf("Enter m, n ");
    scanf("%d %d", &m, &n);

    if (m < n)
        printf("GCD(%d, %d) = %d\n", m, n, GCD(n, m));
    else
        printf("GCD(%d, %d) = %d\n", m, n, GCD(m, n));
}
```

Examples of Recursive Functions

GCD: Dijkstra' Method

```
#include <stdio.h>
int GCD(int m, int n) {
    if (m == n)
        return m;
    if (m > n)
        return GCD(m-n, n);
    return GCD(m, n-m);
}

int main() {
    int m, n;
    printf("Enter m and n: ");
    scanf("%d %d", &m, &n);
    printf("GCD(%d, %d) = %d\n", m, n, GCD(m,n));
}
```

Examples of Recursive Functions

Binomial Coefficient

$$\binom{n}{r} = \begin{cases} 1, \text{ if } r = 0 \\ 1, \text{ if } n = r \\ \binom{n-1}{r} + \binom{n-1}{r-1} \text{ otherwise} \end{cases}$$

Examples of Recursive Functions

Binomial Coefficient

```
#include <stdio.h>
int binom(int n, int r) {
    if (r == 0 || n == r)
        return 1;
    return binom(n-1, r) + binom(n-1, r-1);
}
int main() {
   int n, r;
   printf("Enter n, r: ");
   scanf("%d %d", &n, &r);
   printf("binom(%d, %d) = %d\n", n, r, binom(n, r));
}
```

Examples of Recursive Functions

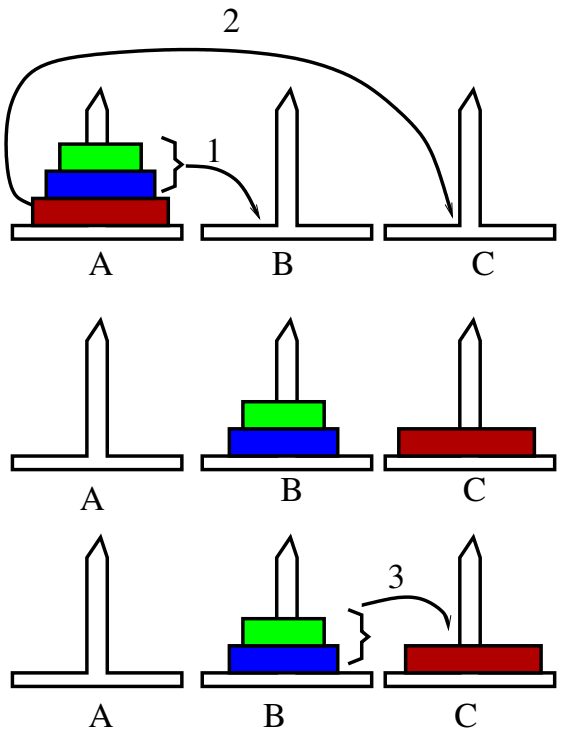
Pascal's Triangle

```
int main() {
    int i, j, n;
    printf("Enter n: ");
    scanf("%d", &n);

    for(i = 0; i < n; i++) {
        for (j = 0; j <= i; j++) {
            printf("%6d", binom(i, j));
        }
        printf("\n");
    }
}
```

Examples of Recursive Functions

Tower of Hanoi



- Two recursive problems of size $n-1$ to be solved.
- Base case is moving the disk with largest diameter.
- So, spec of `tower(n, A, B, C)`:
  - If n $=$ 1 then
    `move disk n from A to C`
  - Else execute following steps:
    1. `tower(n-1, A, C, B)`,
    2. `move disk n from A to C`,
    3. `tower(n-1, B, A, C)`

Examples of Recursive Functions

Tower of Hanoi

```
#include <stdio.h>
void tower(int n, char A, char B, char C) {
    if (n == 1) {
        printf("Move disk 1 from %c to %c\n", A, C);
        return;
    }
    tower(n-1, A, C, B);
    printf("Move disk %d from %c to %c\n", n, A, C);
    tower(n-1, B, A, C);
    return;
}
int main() {
    int n;
    printf("Enter n : ");
    scanf("%d", &n);
    tower(n, 'A', 'B', 'C');
}
```

Merge Sort

Merging