

## Contents

<b>1</b>	<b>Template</b>	1
1.1	Template	1
<b>2</b>	<b>Graph</b>	1
2.1	BFS Algorithm	1
2.2	DFS Algorithm	1
2.3	FloodFill Algorithm	1
2.4	Dijkstra's Algorithm	1
2.5	Floyd Warshall's Algorithm	2
2.6	MST (Kruskal's Algorithm)	2
2.7	Union Find Structure	2
<b>3</b>	<b>DFS</b>	2
3.1	Coin Change	2
3.2	Knapsack	2
3.3	Longest Common Subsequence	2
3.4	Longest Increasing Subsequences	3
<b>4</b>	<b>Query</b>	3
4.1	Prefix sum	3
4.2	Prefix sum 2D	3
4.3	Fenwick Tree	3
4.4	Fenwick Tree 2D	3
4.5	General Segtree	3
4.6	Sum Lazytree	4
<b>5</b>	<b>Strings</b>	4
5.1	KMP	4
5.2	Algorithm Z	4
5.3	Rabin Karp	5
5.4	Aho-Corasick	5
5.5	Hashing	5
5.6	Manacher	6
5.7	Suffix Array	6

## 1 Template

### 1.1 Template

```
#include <bits/stdc++.h>

#define mp make_pair
#define pb push_back
#define ppb pop_back
#define all(a) (a).begin(), (a).end()
#define sz(a) (int)a.size()
#define f first
#define s second
#define forn(i, n) for (int i = 0; i < n; i++)
#define forx(i, x, n) for (int i = x; i < n; i++)
#define each(a, x) for (auto &(a) : (x))

using namespace std;

typedef long long ll;
typedef vector<int> vi;
typedef vector<ll> vl;

void solve() {
    // code here
}

int main()
{
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    solve();
}
```

```
return 0;
}
```

## 2 Graph

### 2.1 BFS Algorithm

```
vector<int> bfs(vector<vector<int>>& g, int v) {
    vector<int> dis(g.size(), -1);
    queue<int> q;
    dis[v] = 0;
    q.push(v);
    while(!q.empty()) {
        int node= q.front();
        q.pop();
        for(int x : g[node]) {
            if(dis[x] == -1) {
                dis[x] = dis[node] + 1;
                q.push(x);
            }
        }
    }
    return dis;
}
```

### 2.2 DFS Algorithm

```
vector<bool> vis(tam);

void dfs(int node){
    vis[node] = 1;
    for(int x : g[node])
        if(!vis[x])
            dfs(x);
}
```

### 2.3 FloodFill Algorithm

```
int n, m;
int dir[2][4] = {{0,0,1,-1}, {1,-1,0,0}};

vector<vector<int>> tab, visi;

int floodfill(int x, int y) {
    if(x < 0 || y < 0 || x >= n || y >= m || visi[x][y] || tab[x][y] == 0)
        return;
    visi[x][y] = 1;
    int ret = 1;
    for(int i = 0; i < 4; i++)
        ret += floodfill(x + dir[0][i], y + dir[1][i]);
    return ret;
}
```

### 2.4 Dijkstra's Algorithm

```
typedef long long ll;
const long long INF = 4e18;

vector<ll> dijkstra(vector<vector<pair<ll, ll>>> graph, int n, int initial_node)
{
    vector<ll> dis(n + 1, INF);
    dis[initial_node] = 0;

    priority_queue<pair<ll, ll>, vector<pair<ll, ll>>, greater<pair<ll, ll>>> pq;
    pq.push({0, initial_node});
    while (!pq.empty())
```

```

{
    pll minor = pq.top();
    pq.pop();
    ll actual_cost = minor.f;
    int node = minor.s;
    if (dis[node] < actual_cost)
        continue;

    for (auto to : graph[node])
    {
        int neighbor = to.f;
        ll cost = to.s;
        if (dis[node] + cost < dis[neighbor])
        {
            dis[neighbor] = dis[node] + cost;
            pq.push({dis[neighbor], neighbor});
        }
    }
}

return dis;
}

```

## 2.5 Floyd Warshall's Algorithm

```

typedef long long ll;

vector<vector<ll>> floydWarshall(vector<vector<pair<ll, ll>> graph, int n)
{
    vector<vector<ll>> dis(n + 1, vl(n + 1, INF));
    forn(i, n) dis[i][i] = 0;

    forn(u, n)
    {
        for (auto to : graph[u])
        {
            ll v = to.f, w = to.s;
            dis[u][v] = min(dis[u][v], w);
            dis[v][u] = min(dis[v][u], w);
        }
    }

    forn(k, n)
    {
        forn(u, n)
        {
            forn(v, n) dis[u][v] = min(dis[u][v], dis[u][k] + dis[k][v]);
        }
    }

    return dis;
}

```

## 2.6 MST (Kruskal's Algorithm)

```

typedef long long ll;

ll kruskal(vector<pair<ll, pair<int, int>>> edges, int n)
{
    sort(all(edges));
    UnionFind dsu(n + 1);
    int countEdges = 0;
    ll res = 0;
    for (auto edge : edges)
    {
        ll weight = edge.f;
        int u = edge.s.f;
        int v = edge.s.s;
        if (dsu.join(u, v))
        {
            countEdges++;
            res += weight;
        }
    }

    if (countEdges == n - 1)
        return res;

    if (countEdges < n - 1)
        return -1;

    return res;
}

```

## 2.7 Union Find Structure

```

struct UnionFind
{
    vector<int> p;
    UnionFind(int n) : p(n, -1) {}

    int find(int x)
    {
        if (p[x] == -1)
            return x;

        return p[x] = find(p[x]);
    }

    bool join(int x, int y)
    {
        x = find(x), y = find(y);
        if (x == y)
            return 0;

        p[y] = x;
        return 1;
    }
};

```

## 3 DFS

### 3.1 Coin Change

```

void solve() {
    ll n_coins, total;
    cin >> n_coins >> total;
    vl dp(total + 1, INT32_MAX - 1);
    vl coins(n_coins);
    forn(i, n_coins) cin >> coins[i];

    dp[0] = 0;
    for(i, n_coins) {
        each(coin, coins) {
            if (coin + i > x) continue;
            dp[coin + i] = min(dp[coin + i], dp[i] + 1);
        }
    }

    if (dp[total] + 1 == INT32_MAX) cout << "-1\n";
    else cout << dp[total] << '\n';
}

```

### 3.2 Knapsack

```

ll knapsack(ll W, vi weights, vi profits, int n) {
    vector<vi> dp(n + 1, vi(W + 1));
    forn(i, n + 1) {
        forn(w, W + 1) {
            if (i == 0 || w == 0) dp[i][w] = 0;
            else if (weights[i - 1] <= w)
                dp[i][w] = max(
                    profit[i - 1] + dp[i - 1][w - weights[i - 1]],
                    dp[i - 1][w]);
            else
                dp[i][w] = dp[i - 1][w];
        }
    }

    return dp[n][W];
}

```

### 3.3 Longest Common Subsequence

```

int lcs(string &s1, string &s2) {
    int m = sz(s1), n = sz(s2);

    vector<vi> dp(m + 1, vi(n + 1, 0));
    forx(i, 1, m + 1) {
        forx(j, 1, n + 1) {
            dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            if (s1[i - 1] == s2[j - 1]) dp[i][j] = max(dp[i][j], dp[i - 1][j - 1] + 1);
        }
    }

    return dp[m][n];
}

```

## 3.4 Longest Increasing Subsequences

```

int lis(vi &original) {
    vi aux;
    forn(i, sz(original)) {
        auto it = lower_bound(all(aux), original[i]);
        if (it == aux.end()) aux.pb(original[i]);
        else *it = original[i];
    }

    return sz(aux);
}

```

# 4 Query

## 4.1 Prefix sum

```

void solve() {
    ll n, q, x, y;
    cin >> n >> q;

    vl nums(n), prefix(n + 1);
    forn(i, n) cin >> nums[i], prefix[i + 1] = prefix[i] + nums[i];

    forn(i, q) {
        cin >> x >> y;
        cout << prefix[y] - prefix[x - 1] << '\n';
    }
}

```

## 4.2 Prefix sum 2D

```

void solve() {
    ll n, q;
    cin >> n >> q;
    vector<string> s(n); // 0-index

    vector<vl> prefix(n + 1, vl(n + 1)); // 1-index
    forn(i, n) {
        forn(j, n) {
            ll value = s[i][j] == '*' ? 1 : 0;
            prefix[i + 1][j + 1] = (value
                                   + prefix[i][j + 1]
                                   + prefix[i + 1][j]
                                   - prefix[i][j]);
        }
    }

    while (q--) {
        ll x1, y1, x2, y2;
        cin >> x1 >> y1 >> x2 >> y2;
        x1--, y1--, x2--, y2--;

        ll sum = (prefix[x2 + 1][y2 + 1]
                  - prefix[x1][y2 + 1]
                  - prefix[x2 + 1][y1]
                  + prefix[x1][y1]); // 0-index query
        cout << sum << '\n';
    }
}

```

## 4.3 Fenwick Tree

```

struct BIT { // 1-index
    vl bit;
    ll n;

    BIT(int n) : bit(n + 1), n(n) {}

    ll lsb(int i) { return i & -i; }

    void add(int i, ll x) {
        for (; i <= n; i += lsb(i)) bit[i] += x;
    }

    ll sum(int r) {
        ll res = 0;
        for (; r > 0; r -= lsb(r)) res += bit[r];
        return res;
    }

    ll sum(int l, int r) {
        return sum(r) - sum(l - 1);
    }

    void set(int i, ll x) {
        add(i, x - sum(i, i));
    }
};

```

## 4.4 Fenwick Tree 2D

```

struct BIT2D {
    vector<vl> bit;
    ll n, m;

    BIT2D(ll n, ll m) : bit(n + 1, vector<ll>(m + 1)), n(n), m(m) {}

    ll lsb(ll i) {
        return i & -i;
    }

    void add(int row, int col, ll x) {
        for (int i = row; i <= n; i += lsb(i)) {
            for (int j = col; j <= m; j += lsb(j)) {
                bit[i][j] += x;
            }
        }
    }

    ll sum(int row, int col) {
        ll res = 0;
        for (int i = row; i > 0; i -= lsb(i)) {
            for (int j = col; j > 0; j -= lsb(j)) {
                res += bit[i][j];
            }
        }

        return res;
    }

    ll sum(int x1, int y1, int x2, int y2) {
        return (sum(x2, y2)
                - sum(x1 - 1, y2)
                - sum(x2, y1 - 1)
                + sum(x1 - 1, y1 - 1));
    }

    void set(int x, int y, ll val) {
        add(x, y, val - sum(x, y, x, y));
    }
};

```

## 4.5 General Segtree

```

struct Node {
    ll a = 0;

    Node(ll val = 0) : a(val) {}
};

```

```

Node e() {
    Node node;
    return node;
}

Node op(Node a, Node b) {
    Node node;
    node.a = a.a ^ b.a;
    return node;
}

struct Segtree {
    vector<Node> nodes;
    ll n;

    void init(int n) {
        auto a = vector<Node>(n, e());
        init(a);
    }

    void init(vector<Node>& initial) {
        nodes.clear();
        n = initial.size();
        int size = 1;
        while (size < n) {
            size *= 2;
        }
        nodes.resize(size * 2);
        build(0, 0, n-1, initial);
    }

    void build(int i, int sl, int sr, vector<Node>& initial) {
        if (sl == sr) {
            nodes[i] = initial[sl];
        } else {
            ll mid = (sl + sr) >> 1;
            build(i*2+1, sl, mid, initial);
            build(i*2+2, mid+1, sr, initial);
            nodes[i] = op(nodes[i*2+1], nodes[i*2+2]);
        }
    }

    void update(int i, int sl, int sr, int pos, Node node) {
        if (sl <= pos && pos <= sr) {
            if (sl == sr) {
                nodes[i] = node;
            } else {
                int mid = (sl + sr) >> 1;
                update(i * 2 + 1, sl, mid, pos, node);
                update(i * 2 + 2, mid + 1, sr, pos, node);
                nodes[i] = op(nodes[i*2+1], nodes[i*2+2]);
            }
        }
    }

    void update(int pos, Node node) {
        update(0, 0, n - 1, pos, node);
    }

    Node query(int i, int sl, int sr, int l, int r) {
        if (l <= sl && sr <= r) {
            return nodes[i];
        } else if (sr < l || r < sl) {
            return e();
        } else {
            int mid = (sl + sr) / 2;
            auto a = query(i * 2 + 1, sl, mid, l, r);
            auto b = query(i * 2 + 2, mid + 1, sr, l, r);
            return op(a, b);
        }
    }

    Node query(int l, int r) {
        return query(0, 0, n - 1, l, r);
    }

    Node get(int i) {
        return query(i, i);
    }
};

```

## 4.6 Sum Lazytree

```

// 0-index
struct Lazytree {
    int n;
    vl sum;

```

```

    vl lazySum;

    void init(int nn) {
        sum.clear();
        n = nn;
        int size = 1;
        while (size < n)
            size *= 2;
        sum.resize(size * 2);
        lazySum.resize(size * 2);
    }

    void update(int i, int sl, int sr, int l, int r, ll diff) {
        if (lazySum[i]) {
            sum[i] += (sr - sl + 1) * lazySum[i];
            if (sl != sr) {
                lazySum[i * 2 + 1] += lazySum[i];
                lazySum[i * 2 + 2] += lazySum[i];
            }
            lazySum[i] = 0;
        }

        if (l <= sl && sr <= r) {
            sum[i] += (sr - sl + 1) * diff;
            if (sl != sr) {
                lazySum[i * 2 + 1] += diff;
                lazySum[i * 2 + 2] += diff;
            }
        } else if (sr < l || r < sl) {
        } else {
            int mid = (sl + sr) >> 1;
            update(i * 2 + 1, sl, mid, l, r, diff);
            update(i * 2 + 2, mid + 1, sr, l, r, diff);
            sum[i] = sum[i * 2 + 1] + sum[i * 2 + 2];
        }
    }

    void update(int l, int r, ll diff) {
        assert(l <= r);
        assert(r < n);
        update(0, 0, n - 1, l, r, diff);
    }

    ll query(int i, int sl, int sr, int l, int r) {
        if (lazySum[i]) {
            sum[i] += lazySum[i] * (sr - sl + 1);
            if (sl != sr) {
                lazySum[i * 2 + 1] += lazySum[i];
                lazySum[i * 2 + 2] += lazySum[i];
            }
            lazySum[i] = 0;
        }

        if (l <= sl && sr <= r) {
            return sum[i];
        } else if (sr < l || r < sl) {
            return 0;
        } else {
            int mid = (sl + sr) >> 1;
            return query(i * 2 + 1, sl, mid, l, r) + query(i * 2 + 2, mid + 1, sr, l, r);
        }
    }

    ll query(int l, int r)
    {
        assert(l <= r);
        assert(r < n);
        return query(0, 0, n - 1, l, r);
    }
};

```

## 5 Strings

### 5.1 KMP

```

vi kmp_builder(string &s, int n) {
    vi dp(n, 0);
    int j = 0;
    forx(i, 1, n) {
        while (j && s[i] != s[j]) j = dp[j - 1];

```

```

    if (s[i] == s[j]) dp[i] = ++j;
    else dp[i] = 0;
}

return dp;
}

// Return all occurrences of the pattern in the text
vi kmp(string &t, string &p) {
    string q = p + "*" + t;
    vi v = kmp_builder(q, sz(q));
    vi res;
    forn(1, sz(q)) if (v[i] == sz(p)) res.pb(i - 2 * sz(p) + 1);

    return res;
}

```

## 5.2 Algorithm Z

```

// Example answer aabb#aaxnaabba -> 01000210041001
vi alz(const string &s) // pattern#where_to_look
{
    int n = s.size();
    vi z(n, 0);
    for(int i = 1, l = 0, r = 0; i < n; i++)
    {
        if(i <= r)
            z[i] = min(z[i - l], r - i + 1);
        while(i + z[i] < n && s[z[i]] == s[i + z[i]])
            z[i]++;
        if(r < i + z[i] - 1)
            l = i, r = i + z[i] - 1;
    }
    return z;
}

```

## 5.3 Rabin Karp

```

const ll mod[2] = {1000000007, 998244353};
const ll px[2] = {29, 31};

vl rabin_karp(string &s, string &p) {
    vl ss[2], pp[2], ppx[2];
    for (ll i = 0; i < 2; i++)
        ss[i] = rolling_hash(s, px[i], mod[i]),
        pp[i] = rolling_hash(p, px[i], mod[i]);

    vi res;
    for (int i = 0; i + sz(p) - 1 < sz(s); i++) {
        ll ok = 1;
        for (ll j = 0; j < 2; j++) {
            int fh = fast_hash(ss[j], px[j], mod[j], i, i + sz(p) - 1) % mod[j];
            ok &= (fh == pp[j].back());
        }
        if (ok) res.pb(i + 1);
    }

    return res;
}

```

## 5.4 Aho-Corasick

```

const int K = 26;

struct Vertex {
    int next[K];
    bool output = false;
    int p = -1;
    char pch;
    int link = -1;
    int go[K];

    Vertex(int p=-1, char ch='s') : p(p), pch(ch) {
        fill(begin(next), end(next), -1);
        fill(begin(go), end(go), -1);
    }
};

```

```

vector<Vertex> t(1);

void aho_init() {
    t.clear();
    t.pb(Vertex());
}

void add_string(string const& s) {
    int v = 0;
    for (char ch : s) {
        int c = ch - 'a';
        if (t[v].next[c] == -1) {
            t[v].next[c] = t.size();
            t.emplace_back(v, ch);
        }
        v = t[v].next[c];
    }
    t[v].output = true;
}

int go(int v, char ch);

int get_link(int v) {
    if (t[v].link == -1) {
        if (v == 0 || t[v].p == 0)
            t[v].link = 0;
        else
            t[v].link = go(get_link(t[v].p), t[v].pch);
    }
    return t[v].link;
}

int go(int v, char ch) {
    int c = ch - 'a';
    if (t[v].go[c] == -1) {
        if (t[v].next[c] != -1)
            t[v].go[c] = t[v].next[c];
        else
            t[v].go[c] = v == 0 ? 0 : go(get_link(v), ch);
    }
    return t[v].go[c];
}

vector<int> search_in_text(const string& text) {
    vector<int> occurrences;
    int v = 0;
    for (int i = 0; i < text.size(); i++) {
        char ch = text[i];
        v = go(v, ch);

        for (int u = v; u != 0; u = get_link(u)) {
            if (t[u].output) {
                occurrences.push_back(i);
            }
        }
    }

    return occurrences;
}

```

## 5.5 Hashing

```

const int K = 2;
struct Hash {
    const ll MOD[K] = {999727999, 1070777777};
    const ll P = 17777771;
    vector<ll> h[K], p[K];
    Hash(string &s) {
        int n = s.size();
        for(int k = 0; k < K; k++) {
            h[k].resize(n + 1, 0);
            p[k].resize(n + 1, 1);
            for(int i = 1; i <= n; i++) {
                h[k][i] = (h[k][i - 1] * P + s[i - 1]) % MOD[k];
                p[k][i] = (p[k][i - 1] * P) % MOD[k];
            }
        }
    }
    vector<ll> get(int i, int j) { // hash [i, j]
        j++;
        vector<ll> r(K);
        for(int k = 0; k < K; k++) {
            r[k] = (h[k][j] - h[k][i] * p[k][j - i]) % MOD[k];
            r[k] = (r[k] + MOD[k]) % MOD[k];
        }
        return r;
    }
};

```

```

};

// Other
ll pow(ll b, ll e, ll m) {
    ll res = 1;
    for (; e >= 1; b = (b * b) % m)
        if (e & 1) res = (res * b) % m;
    return res;
}

ll inv(ll b, ll e, ll m) {
    return pow(pow(b, e, m), m - 2, m);
}

vl rolling_hash(string &s, ll p, ll m) {
    ll n = sz(s);
    vl v(n, 0);
    v[0] = (s[0]) % m;
    for (ll i = 1; i < n; i++)
        v[i] = (v[i - 1] + (s[i] * pow(p, i, m)) % m) % m;

    return v;
}

ll fast_hash(vl &v, ll p, ll m, ll i, ll j) {
    return ((v[j] - (i ? v[i - 1] : 0) * m) % m) * inv(p, i, m) % m;
}

// Hash 128
#define bint __int128
struct Hash {
    bint MOD=212345678987654321LL,P=1777771,PI=106955741089659571LL;
    vector<bint> h,pi;
    Hash(string& s){
        assert((P*PI)%MOD==1);
        h.resize(s.size()+1);pi.resize(s.size()+1);
        h[0]=0;pi[0]=1;
        bint p=1;
        forx(i,l,s.size()+1){
            h[i]=(h[i-1]+p*s[i-1])%MOD;
            pi[i]=(pi[i-1]*PI)%MOD;
            p=(p*P)%MOD;
        }
    }
    ll get(int s, int e){
        return (((h[e]-h[s]+MOD)%MOD)*pi[s])%MOD;
    }
};

```

## 5.6 Manacher

```

/* Find palindromes in a string
f = 1 para pares, 0 impar
a a a a a
1 2 3 3 2 1   f = 0 impar
0 1 2 3 2 1   f = 1 par centrado entre [i-1,i]
Time: O(n)
*/
void manacher(string &s, int f, vi &d) {
    int l = 0, r = -1, n = s.size();
    d.assign(n, 0);
    for (int i = 0; i < n; i++) {
        int k = (i > r ? (1 - f) : min(d[l + r - i + f], r - i + f)) + f;
        while (i + k - f < n && i - k >= 0 && s[i + k - f] == s[i - k]) ++k;
        d[i] = k - f; --k;
        if (i + k - f > r) l = i - k, r = i + k - f;
    }
}

```

## 5.7 Suffix Array

```

struct suffix {
    int index;
    int rank[2];
};

int cmp(struct suffix a, struct suffix b) {
    return (a.rank[0] == b.rank[0])? (a.rank[1] < b.rank[1] ?1: 0):
        (a.rank[0] < b.rank[0] ?1: 0);
}

int *buildSuffixArray(char *txt, int n) {
    struct suffix suffixes[n];

    for (int i = 0; i < n; i++) {
        suffixes[i].index = i;
        suffixes[i].rank[0] = txt[i] - 'a';
        suffixes[i].rank[1] = ((i+1) < n)? (txt[i + 1] - 'a'): -1;
    }

    sort(suffixes, suffixes+n, cmp);

    int ind[n];
    for (int k = 4; k < 2*n; k = k*2) {
        int rank = 0;
        int prev_rank = suffixes[0].rank[0];
        suffixes[0].rank[0] = rank;
        ind[suffixes[0].index] = 0;

        for (int i = 1; i < n; i++) {
            if (suffixes[i].rank[0] == prev_rank &&
                suffixes[i].rank[1] == suffixes[i-1].rank[1]) {
                prev_rank = suffixes[i].rank[0];
                suffixes[i].rank[0] = rank;
            } else {
                prev_rank = suffixes[i].rank[0];
                suffixes[i].rank[0] = ++rank;
            }
            ind[suffixes[i].index] = i;
        }

        for (int i = 0; i < n; i++) {
            int nextindex = suffixes[i].index + k/2;
            suffixes[i].rank[1] = (nextindex < n)?
                suffixes[ind[nextindex]].rank[0]: -1;
        }

        sort(suffixes, suffixes+n, cmp);
    }

    int *suffixArr = new int[n];
    for (int i = 0; i < n; i++)
        suffixArr[i] = suffixes[i].index;

    return suffixArr;
}

void printArr(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}

// Driver program to test above functions
void solve() {
    char txt[] = "banana";
    int n = strlen(txt);
    int *suffixArr = buildSuffixArray(txt, n);
    cout << "Following is suffix array for " << txt << endl;
    printArr(suffixArr, n);
}

```

$f(n) = O(g(n))$	iff $\exists$ positive $c, n_0$ such that $0 \leq f(n) \leq cg(n) \forall n \geq n_0$ .	$\sum_{i=1}^n i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4}.$
$f(n) = \Omega(g(n))$	iff $\exists$ positive $c, n_0$ such that $f(n) \geq cg(n) \geq 0 \forall n \geq n_0$ .	In general:
$f(n) = \Theta(g(n))$	iff $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ .	$\sum_{i=1}^n i^m = \frac{1}{m+1} \left[ (n+1)^{m+1} - 1 - \sum_{i=1}^n ((i+1)^{m+1} - i^{m+1} - (m+1)i^m) \right]$
$f(n) = o(g(n))$	iff $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$ .	$\sum_{i=1}^{n-1} i^m = \frac{1}{m+1} \sum_{k=0}^m \binom{m+1}{k} B_k n^{m+1-k}.$
$\lim_{n \rightarrow \infty} a_n = a$	iff $\forall \epsilon > 0, \exists n_0$ such that $ a_n - a  < \epsilon, \forall n \geq n_0$ .	Geometric series:
$\sup S$	least $b \in \mathbb{R}$ such that $b \geq s, \forall s \in S$ .	$\sum_{i=0}^n c^i = \frac{c^{n+1} - 1}{c - 1}, \quad c \neq 1, \quad \sum_{i=0}^{\infty} c^i = \frac{1}{1 - c}, \quad \sum_{i=1}^{\infty} c^i = \frac{c}{1 - c}, \quad  c  < 1,$
$\inf S$	greatest $b \in \mathbb{R}$ such that $b \leq s, \forall s \in S$ .	$\sum_{i=0}^n i c^i = \frac{nc^{n+2} - (n+1)c^{n+1} + c}{(c-1)^2}, \quad c \neq 1, \quad \sum_{i=0}^{\infty} i c^i = \frac{c}{(1-c)^2}, \quad  c  < 1.$
$\liminf_{n \rightarrow \infty} a_n$	$\lim_{n \rightarrow \infty} \inf \{a_i \mid i \geq n, i \in \mathbb{N}\}.$	Harmonic series:
$\limsup_{n \rightarrow \infty} a_n$	$\lim_{n \rightarrow \infty} \sup \{a_i \mid i \geq n, i \in \mathbb{N}\}.$	$H_n = \sum_{i=1}^n \frac{1}{i}, \quad \sum_{i=1}^n i H_i = \frac{n(n+1)}{2} H_n - \frac{n(n-1)}{4}.$
$\binom{n}{k}$	Combinations: Size $k$ sub-sets of a size $n$ set.	$\sum_{i=1}^n H_i = (n+1)H_n - n, \quad \sum_{i=1}^n \binom{i}{m} H_i = \binom{n+1}{m+1} \left( H_{n+1} - \frac{1}{m+1} \right).$
$[n]$	Stirling numbers (1st kind): Arrangements of an $n$ element set into $k$ cycles.	1. $\binom{n}{k} = \frac{n!}{(n-k)!k!}, \quad 2. \sum_{k=0}^n \binom{n}{k} = 2^n, \quad 3. \binom{n}{k} = \binom{n}{n-k},$
$\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$	Stirling numbers (2nd kind): Partitions of an $n$ element set into $k$ non-empty sets.	4. $\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1}, \quad 5. \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1},$
$\langle \begin{matrix} n \\ k \end{matrix} \rangle$	1st order Eulerian numbers: Permutations $\pi_1 \pi_2 \dots \pi_n$ on $\{1, 2, \dots, n\}$ with $k$ ascents.	6. $\binom{n}{m} \binom{m}{k} = \binom{n}{k} \binom{n-k}{m-k}, \quad 7. \sum_{k=0}^n \binom{r+k}{k} = \binom{r+n+1}{n},$
$\langle \langle \begin{matrix} n \\ k \end{matrix} \rangle \rangle$	2nd order Eulerian numbers.	8. $\sum_{k=0}^n \binom{k}{m} = \binom{n+1}{m+1}, \quad 9. \sum_{k=0}^n \binom{r}{k} \binom{s}{n-k} = \binom{r+s}{n},$
$C_n$	Catalan Numbers: Binary trees with $n+1$ vertices.	10. $\binom{n}{k} = (-1)^k \binom{k-n-1}{k}, \quad 11. \left\{ \begin{matrix} n \\ 1 \end{matrix} \right\} = \left\{ \begin{matrix} n \\ n \end{matrix} \right\} = 1,$
14. $\left[ \begin{matrix} n \\ 1 \end{matrix} \right] = (n-1)!,$	15. $\left[ \begin{matrix} n \\ 2 \end{matrix} \right] = (n-1)!H_{n-1},$	16. $\left[ \begin{matrix} n \\ n \end{matrix} \right] = 1, \quad 17. \left[ \begin{matrix} n \\ k \end{matrix} \right] \geq \left\{ \begin{matrix} n \\ k \end{matrix} \right\},$
18. $\left[ \begin{matrix} n \\ k \end{matrix} \right] = (n-1) \left[ \begin{matrix} n-1 \\ k \end{matrix} \right] + \left[ \begin{matrix} n-1 \\ k-1 \end{matrix} \right],$	19. $\left\{ \begin{matrix} n \\ n-1 \end{matrix} \right\} = \left[ \begin{matrix} n \\ n-1 \end{matrix} \right] = \binom{n}{2},$	20. $\sum_{k=0}^n \left[ \begin{matrix} n \\ k \end{matrix} \right] = n!, \quad 21. C_n = \frac{1}{n+1} \binom{2n}{n},$
22. $\langle \begin{matrix} n \\ 0 \end{matrix} \rangle = \langle \begin{matrix} n \\ n-1 \end{matrix} \rangle = 1,$	23. $\langle \begin{matrix} n \\ k \end{matrix} \rangle = \langle \begin{matrix} n \\ n-1-k \end{matrix} \rangle,$	24. $\langle \begin{matrix} n \\ k \end{matrix} \rangle = (k+1) \langle \begin{matrix} n-1 \\ k \end{matrix} \rangle + (n-k) \langle \begin{matrix} n-1 \\ k-1 \end{matrix} \rangle,$
25. $\langle \begin{matrix} 0 \\ k \end{matrix} \rangle = \begin{cases} 1 & \text{if } k=0, \\ 0 & \text{otherwise} \end{cases}$	26. $\langle \begin{matrix} n \\ 1 \end{matrix} \rangle = 2^n - n - 1,$	27. $\langle \begin{matrix} n \\ 2 \end{matrix} \rangle = 3^n - (n+1)2^n + \binom{n+1}{2},$
28. $x^n = \sum_{k=0}^n \langle \begin{matrix} n \\ k \end{matrix} \rangle \binom{x+k}{n},$	29. $\langle \begin{matrix} n \\ m \end{matrix} \rangle = \sum_{k=0}^m \binom{n+1}{k} (m+1-k)^n (-1)^k,$	30. $m! \left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \sum_{k=0}^n \langle \begin{matrix} n \\ k \end{matrix} \rangle \binom{k}{n-m},$
31. $\langle \begin{matrix} n \\ m \end{matrix} \rangle = \sum_{k=0}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\} \binom{n-k}{m} (-1)^{n-k-m} k!,$	32. $\langle \langle \begin{matrix} n \\ 0 \end{matrix} \rangle \rangle = 1,$	33. $\langle \langle \begin{matrix} n \\ n \end{matrix} \rangle \rangle = 0 \quad \text{for } n \neq 0,$
34. $\langle \langle \begin{matrix} n \\ k \end{matrix} \rangle \rangle = (k+1) \langle \langle \begin{matrix} n-1 \\ k \end{matrix} \rangle \rangle + (2n-1-k) \langle \langle \begin{matrix} n-1 \\ k-1 \end{matrix} \rangle \rangle,$	35. $\sum_{k=0}^n \langle \langle \begin{matrix} n \\ k \end{matrix} \rangle \rangle = \frac{(2n)n}{2^n},$	
36. $\left\{ \begin{matrix} x \\ x-n \end{matrix} \right\} = \sum_{k=0}^n \langle \langle \begin{matrix} n \\ k \end{matrix} \rangle \rangle \binom{x+n-1-k}{2n},$	37. $\left\{ \begin{matrix} n+1 \\ m+1 \end{matrix} \right\} = \sum_k \binom{n}{k} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} = \sum_{k=0}^n \left\{ \begin{matrix} k \\ m \end{matrix} \right\} (m+1)^{n-k},$	

<p>38. <math>\begin{bmatrix} n+1 \\ m+1 \end{bmatrix} = \sum_k \begin{bmatrix} n \\ k \end{bmatrix} \begin{bmatrix} k \\ m \end{bmatrix} = \sum_{k=0}^n \begin{bmatrix} k \\ m \end{bmatrix} n^{n-k} = n! \sum_{k=0}^n \frac{1}{k!} \begin{bmatrix} k \\ m \end{bmatrix},</math></p> <p>40. <math>\left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \sum_k \binom{n}{k} \left\{ \begin{matrix} k+1 \\ m+1 \end{matrix} \right\} (-1)^{n-k},</math></p> <p>42. <math>\left\{ \begin{matrix} m+n+1 \\ m \end{matrix} \right\} = \sum_{k=0}^m k \left\{ \begin{matrix} n+k \\ k \end{matrix} \right\},</math></p> <p>44. <math>\binom{n}{m} = \sum_k \left\{ \begin{matrix} n+1 \\ k+1 \end{matrix} \right\} \begin{bmatrix} k \\ m \end{bmatrix} (-1)^{m-k},</math></p> <p>46. <math>\left\{ \begin{matrix} n \\ n-m \end{matrix} \right\} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \begin{bmatrix} m+k \\ k \end{bmatrix},</math></p> <p>48. <math>\left\{ \begin{matrix} n \\ \ell+m \end{matrix} \right\} \binom{\ell+m}{\ell} = \sum_k \left\{ \begin{matrix} k \\ \ell \end{matrix} \right\} \left\{ \begin{matrix} n-k \\ m \end{matrix} \right\} \binom{n}{k},</math></p>	<p>39. <math>\begin{bmatrix} x \\ x-n \end{bmatrix} = \sum_{k=0}^n \left\langle \begin{matrix} n \\ k \end{matrix} \right\rangle \begin{bmatrix} x+k \\ 2n \end{bmatrix},</math></p> <p>41. <math>\begin{bmatrix} n \\ m \end{bmatrix} = \sum_k \begin{bmatrix} n+1 \\ k+1 \end{bmatrix} \binom{k}{m} (-1)^{m-k},</math></p> <p>43. <math>\begin{bmatrix} m+n+1 \\ m \end{bmatrix} = \sum_{k=0}^m k(n+k) \begin{bmatrix} n+k \\ k \end{bmatrix},</math></p> <p>45. <math>(n-m)! \binom{n}{m} = \sum_k \begin{bmatrix} n+1 \\ k+1 \end{bmatrix} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} (-1)^{m-k},</math> for <math>n \geq m,</math></p> <p>47. <math>\begin{bmatrix} n \\ n-m \end{bmatrix} = \sum_k \binom{m-n}{m+k} \binom{m+n}{n+k} \left\{ \begin{matrix} m+k \\ k \end{matrix} \right\},</math></p> <p>49. <math>\begin{bmatrix} n \\ \ell+m \end{bmatrix} \binom{\ell+m}{\ell} = \sum_k \begin{bmatrix} k \\ \ell \end{bmatrix} \begin{bmatrix} n-k \\ m \end{bmatrix} \binom{n}{k}.</math></p>	<p>Every tree with <math>n</math> vertices has <math>n-1</math> edges.</p> <p>Kraft inequality: If the depths of the leaves of a binary tree are <math>d_1, \dots, d_n</math>:</p> $\sum_{i=1}^n 2^{-d_i} \leq 1,$ <p>and equality holds only if every internal node has 2 sons.</p>
--	--	--

## Recurrences

Master method:

$$T(n) = aT(n/b) + f(n), \quad a \geq 1, b > 1$$

If  $\exists \epsilon > 0$  such that  $f(n) = O(n^{\log_b a - \epsilon})$  then

$$T(n) = \Theta(n^{\log_b a}).$$

If  $f(n) = \Theta(n^{\log_b a})$  then

$$T(n) = \Theta(n^{\log_b a} \log_2 n).$$

If  $\exists \epsilon > 0$  such that  $f(n) = \Omega(n^{\log_b a + \epsilon})$ , and  $\exists c < 1$  such that  $af(n/b) \leq cf(n)$  for large  $n$ , then

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

Substitution (example): Consider the following recurrence

$$T_{i+1} = 2^{2^i} \cdot T_i^2, \quad T_1 = 2.$$

Note that  $T_i$  is always a power of two.

Let  $t_i = \log_2 T_i$ . Then we have

$$t_{i+1} = 2^i + 2t_i, \quad t_1 = 1.$$

Let  $u_i = t_i/2^i$ . Dividing both sides of the previous equation by  $2^{i+1}$  we get

$$\frac{t_{i+1}}{2^{i+1}} = \frac{2^i}{2^{i+1}} + \frac{t_i}{2^i}.$$

Substituting we find

$$u_{i+1} = \frac{1}{2} + u_i, \quad u_1 = \frac{1}{2},$$

which is simply  $u_i = i/2$ . So we find that  $T_i$  has the closed form  $T_i = 2^{i2^{i-1}}$ .

Summing factors (example): Consider the following recurrence

$$T(n) = 3T(n/2) + n, \quad T(1) = 1.$$

Rewrite so that all terms involving  $T$  are on the left side

$$T(n) - 3T(n/2) = n.$$

Now expand the recurrence, and choose a factor which makes the left side “telescope”

$$1(T(n) - 3T(n/2)) = n$$

$$3(T(n/2) - 3T(n/4)) = n/2$$

$$\vdots \quad \vdots \quad \vdots$$

$$3^{\log_2 n - 1} (T(2) - 3T(1)) = 2$$

Let  $m = \log_2 n$ . Summing the left side we get  $T(n) - 3^m T(1) = T(n) - 3^m = T(n) - n^k$  where  $k = \log_2 3 \approx 1.58496$ . Summing the right side we get

$$\sum_{i=0}^{m-1} \frac{n}{2^i} 3^i = n \sum_{i=0}^{m-1} \left(\frac{3}{2}\right)^i.$$

Let  $c = \frac{3}{2}$ . Then we have

$$n \sum_{i=0}^{m-1} c^i = n \left( \frac{c^m - 1}{c - 1} \right)$$

$$= 2n(c^{\log_2 n} - 1)$$

$$= 2n(c^{(k-1)\log_2 n} - 1)$$

$$= 2n^k - 2n,$$

and so  $T(n) = 3n^k - 2n$ . Full history recurrences can often be changed to limited history ones (example): Consider

$$T_i = 1 + \sum_{j=0}^{i-1} T_j, \quad T_0 = 1.$$

Note that

$$T_{i+1} = 1 + \sum_{j=0}^i T_j.$$

Subtracting we find

$$\begin{aligned} T_{i+1} - T_i &= 1 + \sum_{j=0}^i T_j - 1 - \sum_{j=0}^{i-1} T_j \\ &= T_i. \end{aligned}$$

And so  $T_{i+1} = 2T_i = 2^{i+1}$ .

Generating functions:

1. Multiply both sides of the equation by  $x^i$ .
2. Sum both sides over all  $i$  for which the equation is valid.
3. Choose a generating function  $G(x)$ . Usually  $G(x) = \sum_{i=0}^{\infty} x^i g_i$ .
3. Rewrite the equation in terms of the generating function  $G(x)$ .
4. Solve for  $G(x)$ .
5. The coefficient of  $x^i$  in  $G(x)$  is  $g_i$ .

Example:

$$g_{i+1} = 2g_i + 1, \quad g_0 = 0.$$

Multiply and sum:

$$\sum_{i \geq 0} g_{i+1} x^i = \sum_{i \geq 0} 2g_i x^i + \sum_{i \geq 0} x^i.$$

We choose  $G(x) = \sum_{i \geq 0} x^i g_i$ . Rewrite in terms of  $G(x)$ :

$$\frac{G(x) - g_0}{x} = 2G(x) + \sum_{i \geq 0} x^i.$$

Simplify:

$$\frac{G(x)}{x} = 2G(x) + \frac{1}{1-x}.$$

Solve for  $G(x)$ :

$$G(x) = \frac{x}{(1-x)(1-2x)}.$$

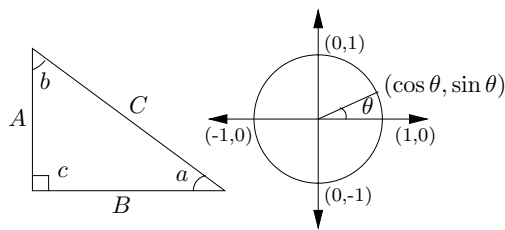
Expand this using partial fractions:

$$\begin{aligned} G(x) &= x \left( \frac{2}{1-2x} - \frac{1}{1-x} \right) \\ &= x \left( 2 \sum_{i \geq 0} 2^i x^i - \sum_{i \geq 0} x^i \right) \\ &= \sum_{i \geq 0} (2^{i+1} - 1) x^{i+1}. \end{aligned}$$

So  $g_i = 2^i - 1$ .



$n \sim 0.11109,$			$\psi - 2 \sim 1.01009,$	$\psi - 2 \sim 0.01009$
$i$	$2^i$	$p_i$	General	Probability
1	2	2	Bernoulli Numbers ( $B_i = 0$ , odd $i \neq 1$ ):	Continuous distributions: If
2	4	3	$B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_4 = -\frac{1}{30},$	$\Pr[a < X < b] = \int_a^b p(x) dx,$
3	8	5	$B_6 = \frac{1}{42}, B_8 = -\frac{1}{30}, B_{10} = \frac{5}{66}.$	then $p$ is the probability density function of $X$ . If
4	16	7	Change of base, quadratic formula:	$\Pr[X < a] = P(a),$
5	32	11	$\log_b x = \frac{\log_a x}{\log_a b}, \quad \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$	then $P$ is the distribution function of $X$ . If $P$ and $p$ both exist then
6	64	13	Euler's number $e$ :	$P(a) = \int_{-\infty}^a p(x) dx.$
7	128	17	$e = 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \dots$	Expectation: If $X$ is discrete
8	256	19	$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x.$	$E[g(X)] = \sum_x g(x) \Pr[X = x].$
9	512	23	$\left(1 + \frac{1}{n}\right)^n < e < \left(1 + \frac{1}{n}\right)^{n+1}.$	If $X$ continuous then
10	1,024	29	$\left(1 + \frac{1}{n}\right)^n = e - \frac{e}{2n} + \frac{11e}{24n^2} - O\left(\frac{1}{n^3}\right).$	$E[g(X)] = \int_{-\infty}^{\infty} g(x)p(x) dx = \int_{-\infty}^{\infty} g(x) dP(x).$
11	2,048	31	Harmonic numbers:	Variance, standard deviation:
12	4,096	37	$1, \frac{3}{2}, \frac{11}{6}, \frac{25}{12}, \frac{137}{60}, \frac{49}{20}, \frac{363}{140}, \frac{761}{280}, \frac{7129}{2520}, \dots$	$\text{VAR}[X] = E[X^2] - E[X]^2,$
13	8,192	41	$\ln n < H_n < \ln n + 1,$	$\sigma = \sqrt{\text{VAR}[X]}.$
14	16,384	43	$H_n = \ln n + \gamma + O\left(\frac{1}{n}\right).$	For events $A$ and $B$ :
15	32,768	47	Factorial, Stirling's approximation:	$\Pr[A \vee B] = \Pr[A] + \Pr[B] - \Pr[A \wedge B]$
16	65,536	53	$1, 2, 6, 24, 120, 720, 5040, 40320, 362880, \dots$	$\Pr[A \wedge B] = \Pr[A] \cdot \Pr[B],$
17	131,072	59	$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right).$	iff $A$ and $B$ are independent.
18	262,144	61	Ackermann's function and inverse:	$\Pr[A B] = \frac{\Pr[A \wedge B]}{\Pr[B]}$
19	524,288	67	$a(i, j) = \begin{cases} 2^j & i = 1 \\ a(i-1, 2) & j = 1 \\ a(i-1, a(i, j-1)) & i, j \geq 2 \end{cases}$	For random variables $X$ and $Y$ :
20	1,048,576	71	$\alpha(i) = \min\{j \mid a(j, j) \geq i\}.$	$E[X \cdot Y] = E[X] \cdot E[Y],$
21	2,097,152	73	Binomial distribution:	if $X$ and $Y$ are independent.
22	4,194,304	79	$\Pr[X = k] = \binom{n}{k} p^k q^{n-k}, \quad q = 1 - p,$	$E[X + Y] = E[X] + E[Y],$
23	8,388,608	83	$E[X] = \sum_{k=1}^n k \binom{n}{k} p^k q^{n-k} = np.$	$E[cX] = cE[X].$
24	16,777,216	89	Poisson distribution:	Bayes' theorem:
25	33,554,432	97	$\Pr[X = k] = \frac{e^{-\lambda} \lambda^k}{k!}, \quad E[X] = \lambda.$	$\Pr[A_i B] = \frac{\Pr[B A_i] \Pr[A_i]}{\sum_{j=1}^n \Pr[A_j] \Pr[B A_j]}.$
26	67,108,864	101	Normal (Gaussian) distribution:	Inclusion-exclusion:
27	134,217,728	103	$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, \quad E[X] = \mu.$	$\Pr\left[\bigvee_{i=1}^n X_i\right] = \sum_{i=1}^n \Pr[X_i] +$
28	268,435,456	107	The "coupon collector": We are given a random coupon each day, and there are $n$ different types of coupons. The distribution of coupons is uniform. The expected number of days to pass before we to collect all $n$ types is	$\sum_{k=2}^n (-1)^{k+1} \sum_{i_1 < \dots < i_k} \Pr\left[\bigwedge_{j=1}^k X_{i_j}\right].$
29	536,870,912	109	$nH_n.$	Moment inequalities:
30	1,073,741,824	113		$\Pr[ X  \geq \lambda E[X]] \leq \frac{1}{\lambda},$
31	2,147,483,648	127		$\Pr[ X - E[X]  \geq \lambda \cdot \sigma] \leq \frac{1}{\lambda^2}.$
32	4,294,967,296	131		Geometric distribution:
Pascal's Triangle				$\Pr[X = k] = pq^{k-1}, \quad q = 1 - p,$
1				$E[X] = \sum_{k=1}^{\infty} kpq^{k-1} = \frac{1}{p}.$
1 1				
1 2 1				
1 3 3 1				
1 4 6 4 1				
1 5 10 10 5 1				
1 6 15 20 15 6 1				
1 7 21 35 35 21 7 1				
1 8 28 56 70 56 28 8 1				
1 9 36 84 126 126 84 36 9 1				
1 10 45 120 210 252 210 120 45 10 1				



Pythagorean theorem:

$$C^2 = A^2 + B^2.$$

Definitions:

$$\begin{aligned} \sin a &= A/C, & \cos a &= B/C, \\ \csc a &= C/A, & \sec a &= C/B, \\ \tan a &= \frac{\sin a}{\cos a} = \frac{A}{B}, & \cot a &= \frac{\cos a}{\sin a} = \frac{B}{A}. \end{aligned}$$

Area, radius of inscribed circle:

$$\frac{1}{2}AB, \quad \frac{AB}{A+B+C}.$$

Identities:

$$\begin{aligned} \sin x &= \frac{1}{\csc x}, & \cos x &= \frac{1}{\sec x}, \\ \tan x &= \frac{1}{\cot x}, & \sin^2 x + \cos^2 x &= 1, \\ 1 + \tan^2 x &= \sec^2 x, & 1 + \cot^2 x &= \csc^2 x, \\ \sin x &= \cos\left(\frac{\pi}{2} - x\right), & \sin x &= \sin(\pi - x), \\ \cos x &= -\cos(\pi - x), & \tan x &= \cot\left(\frac{\pi}{2} - x\right), \\ \cot x &= -\cot(\pi - x), & \csc x &= \cot\frac{\pi}{2} - \cot x, \\ \sin(x \pm y) &= \sin x \cos y \pm \cos x \sin y, \\ \cos(x \pm y) &= \cos x \cos y \mp \sin x \sin y, \\ \tan(x \pm y) &= \frac{\tan x \pm \tan y}{1 \mp \tan x \tan y}, \\ \cot(x \pm y) &= \frac{\cot x \cot y \mp 1}{\cot x \pm \cot y}, \\ \sin 2x &= 2 \sin x \cos x, & \sin 2x &= \frac{2 \tan x}{1 + \tan^2 x}, \\ \cos 2x &= \cos^2 x - \sin^2 x, & \cos 2x &= 2 \cos^2 x - 1, \\ \cos 2x &= 1 - 2 \sin^2 x, & \cos 2x &= \frac{1 - \tan^2 x}{1 + \tan^2 x}, \\ \tan 2x &= \frac{2 \tan x}{1 - \tan^2 x}, & \cot 2x &= \frac{\cot^2 x - 1}{2 \cot x}, \\ \sin(x+y) \sin(x-y) &= \sin^2 x - \sin^2 y, \\ \cos(x+y) \cos(x-y) &= \cos^2 x - \sin^2 y. \end{aligned}$$

Euler's equation:

$$e^{ix} = \cos x + i \sin x, \quad e^{i\pi} = -1.$$

Multiplication:

$$C = A \cdot B, \quad c_{i,j} = \sum_{k=1}^n a_{i,k} b_{k,j}.$$

Determinants:  $\det A \neq 0$  iff  $A$  is non-singular.

$$\det A \cdot B = \det A \cdot \det B,$$

$$\det A = \sum_{\pi} \prod_{i=1}^n \text{sign}(\pi) a_{i,\pi(i)}.$$

$2 \times 2$  and  $3 \times 3$  determinant:

$$\begin{aligned} \begin{vmatrix} a & b \\ c & d \end{vmatrix} &= ad - bc, \\ \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} &= g \begin{vmatrix} b & c \\ e & f \end{vmatrix} - h \begin{vmatrix} a & c \\ d & f \end{vmatrix} + i \begin{vmatrix} a & b \\ d & e \end{vmatrix} \\ &= aei + bfg + cdh \\ &\quad - ceg - fha - ibd. \end{aligned}$$

Permanents:

$$\text{perm } A = \sum_{\pi} \prod_{i=1}^n a_{i,\pi(i)}.$$

### Hyperbolic Functions

Definitions:

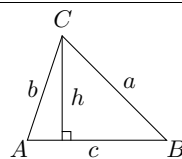
$$\begin{aligned} \sinh x &= \frac{e^x - e^{-x}}{2}, & \cosh x &= \frac{e^x + e^{-x}}{2}, \\ \tanh x &= \frac{e^x - e^{-x}}{e^x + e^{-x}}, & \text{csch } x &= \frac{1}{\sinh x}, \\ \text{sech } x &= \frac{1}{\cosh x}, & \coth x &= \frac{1}{\tanh x}. \end{aligned}$$

Identities:

$$\begin{aligned} \cosh^2 x - \sinh^2 x &= 1, & \tanh^2 x + \text{sech}^2 x &= 1, \\ \coth^2 x - \text{csch}^2 x &= 1, & \sinh(-x) &= -\sinh x, \\ \cosh(-x) &= \cosh x, & \tanh(-x) &= -\tanh x, \\ \sinh(x+y) &= \sinh x \cosh y + \cosh x \sinh y, \\ \cosh(x+y) &= \cosh x \cosh y + \sinh x \sinh y, \\ \sinh 2x &= 2 \sinh x \cosh x, \\ \cosh 2x &= \cosh^2 x + \sinh^2 x, \\ \cosh x + \sinh x &= e^x, & \cosh x - \sinh x &= e^{-x}, \\ (\cosh x + \sinh x)^n &= \cosh nx + \sinh nx, & n \in \mathbb{Z}, \\ 2 \sinh^2 \frac{x}{2} &= \cosh x - 1, & 2 \cosh^2 \frac{x}{2} &= \cosh x + 1. \end{aligned}$$

$\theta$	$\sin \theta$	$\cos \theta$	$\tan \theta$
0	0	1	0
$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	1
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$
$\frac{\pi}{2}$	1	0	$\infty$

... in mathematics  
you don't under-  
stand things, you  
just get used to  
them.  
- J. von Neumann



Law of cosines:

$$c^2 = a^2 + b^2 - 2ab \cos C.$$

Area:

$$\begin{aligned} A &= \frac{1}{2}hc, \\ &= \frac{1}{2}ab \sin C, \\ &= \frac{c^2 \sin A \sin B}{2 \sin C}. \end{aligned}$$

Heron's formula:

$$\begin{aligned} A &= \sqrt{s \cdot s_a \cdot s_b \cdot s_c}, \\ s &= \frac{1}{2}(a + b + c), \\ s_a &= s - a, \\ s_b &= s - b, \\ s_c &= s - c. \end{aligned}$$

More identities:

$$\begin{aligned} \sin \frac{x}{2} &= \sqrt{\frac{1 - \cos x}{2}}, \\ \cos \frac{x}{2} &= \sqrt{\frac{1 + \cos x}{2}}, \\ \tan \frac{x}{2} &= \sqrt{\frac{1 - \cos x}{1 + \cos x}}, \\ &= \frac{1 - \cos x}{\sin x}, \\ &= \frac{\sin x}{1 + \cos x}, \\ \cot \frac{x}{2} &= \sqrt{\frac{1 + \cos x}{1 - \cos x}}, \\ &= \frac{1 + \cos x}{\sin x}, \\ &= \frac{\sin x}{1 - \cos x}, \\ \sin x &= \frac{e^{ix} - e^{-ix}}{2i}, \\ \cos x &= \frac{e^{ix} + e^{-ix}}{2}, \\ \tan x &= -i \frac{e^{ix} - e^{-ix}}{e^{ix} + e^{-ix}}, \\ &= -i \frac{e^{2ix} - 1}{e^{2ix} + 1}, \\ \sin x &= \frac{\sinh ix}{i}, \\ \cos x &= \cosh ix, \\ \tan x &= \frac{\tanh ix}{i}. \end{aligned}$$

The Chinese remainder theorem: There exists a number  $C$  such that:

$$C \equiv r_1 \pmod{m_1}$$

$$\vdots \quad \vdots \quad \vdots$$

$$C \equiv r_n \pmod{m_n}$$

if  $m_i$  and  $m_j$  are relatively prime for  $i \neq j$ .

Euler's function:  $\phi(x)$  is the number of positive integers less than  $x$  relatively prime to  $x$ . If  $\prod_{i=1}^n p_i^{e_i}$  is the prime factorization of  $x$  then

$$\phi(x) = \prod_{i=1}^n p_i^{e_i-1} (p_i - 1).$$

Euler's theorem: If  $a$  and  $b$  are relatively prime then

$$1 \equiv a^{\phi(b)} \pmod{b}.$$

Fermat's theorem:

$$1 \equiv a^{p-1} \pmod{p}.$$

The Euclidean algorithm: if  $a > b$  are integers then

$$\gcd(a, b) = \gcd(a \bmod b, b).$$

If  $\prod_{i=1}^n p_i^{e_i}$  is the prime factorization of  $x$  then

$$S(x) = \sum_{d|x} d = \prod_{i=1}^n \frac{p_i^{e_i+1} - 1}{p_i - 1}.$$

Perfect Numbers:  $x$  is an even perfect number iff  $x = 2^{n-1}(2^n - 1)$  and  $2^n - 1$  is prime.

Wilson's theorem:  $n$  is a prime iff

$$(n-1)! \equiv -1 \pmod{n}.$$

Möbius inversion:

$$\mu(i) = \begin{cases} 1 & \text{if } i = 1. \\ 0 & \text{if } i \text{ is not square-free.} \\ (-1)^r & \text{if } i \text{ is the product of } r \text{ distinct primes.} \end{cases}$$

If

$$G(a) = \sum_{d|a} F(d),$$

then

$$F(a) = \sum_{d|a} \mu(d) G\left(\frac{a}{d}\right).$$

Prime numbers:

$$p_n = n \ln n + n \ln \ln n - n + n \frac{\ln \ln n}{\ln n}$$

$$+ O\left(\frac{n}{\ln n}\right),$$

$$\pi(n) = \frac{n}{\ln n} + \frac{n}{(\ln n)^2} + \frac{2!n}{(\ln n)^3}$$

$$+ O\left(\frac{n}{(\ln n)^4}\right).$$

Definitions:

*Loop* An edge connecting a vertex to itself.

*Directed* Each edge has a direction.

*Simple* Graph with no loops or multi-edges.

*Walk* A sequence  $v_0 e_1 v_1 \dots e_\ell v_\ell$ .

*Trail* A walk with distinct edges.

*Path* A trail with distinct vertices.

*Connected* A graph where there exists a path between any two vertices.

*Component* A maximal connected subgraph.

*Tree* A connected acyclic graph.

*Free tree* A tree with no root.

*DAG* Directed acyclic graph.

*Eulerian* Graph with a trail visiting each edge exactly once.

*Hamiltonian* Graph with a cycle visiting each vertex exactly once.

*Cut* A set of edges whose removal increases the number of components.

*Cut-set* A minimal cut.

*Cut edge* A size 1 cut.

*k-Connected* A graph connected with the removal of any  $k-1$  vertices.

*k-Tough*  $\forall S \subseteq V, S \neq \emptyset$  we have  $k \cdot c(G-S) \leq |S|$ .

*k-Regular* A graph where all vertices have degree  $k$ .

*k-Factor* A  $k$ -regular spanning subgraph.

*Matching* A set of edges, no two of which are adjacent.

*Clique* A set of vertices, all of which are adjacent.

*Ind. set* A set of vertices, none of which are adjacent.

*Vertex cover* A set of vertices which cover all edges.

*Planar graph* A graph which can be embedded in the plane.

*Plane graph* An embedding of a planar graph.

$$\sum_{v \in V} \deg(v) = 2m.$$

If  $G$  is planar then  $n - m + f = 2$ , so

$$f \leq 2n - 4, \quad m \leq 3n - 6.$$

Any planar graph has a vertex with degree  $\leq 5$ .

Notation:

$E(G)$  Edge set

$V(G)$  Vertex set

$c(G)$  Number of components

$G[S]$  Induced subgraph

$\deg(v)$  Degree of  $v$

$\Delta(G)$  Maximum degree

$\delta(G)$  Minimum degree

$\chi(G)$  Chromatic number

$\chi_E(G)$  Edge chromatic number

$G^c$  Complement graph

$K_n$  Complete graph

$K_{n_1, n_2}$  Complete bipartite graph

$r(k, \ell)$  Ramsey number

Geometry

Projective coordinates: triples  $(x, y, z)$ , not all  $x, y$  and  $z$  zero.

$$(x, y, z) = (cx, cy, cz) \quad \forall c \neq 0.$$

Cartesian Projective

$$(x, y) \quad (x, y, 1)$$

$$y = mx + b \quad (m, -1, b)$$

$$x = c \quad (1, 0, -c)$$

Distance formula,  $L_p$  and  $L_\infty$  metric:

$$\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2},$$

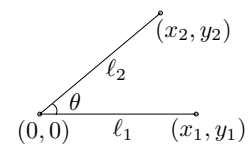
$$[|x_1 - x_0|^p + |y_1 - y_0|^p]^{1/p},$$

$$\lim_{p \rightarrow \infty} [|x_1 - x_0|^p + |y_1 - y_0|^p]^{1/p}.$$

Area of triangle  $(x_0, y_0), (x_1, y_1)$  and  $(x_2, y_2)$ :

$$\frac{1}{2} \text{abs} \begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{vmatrix}.$$

Angle formed by three points:



$$\cos \theta = \frac{(x_1, y_1) \cdot (x_2, y_2)}{l_1 l_2}.$$

Line through two points  $(x_0, y_0)$  and  $(x_1, y_1)$ :

$$\begin{vmatrix} x & y & 1 \\ x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \end{vmatrix} = 0.$$

Area of circle, volume of sphere:

$$A = \pi r^2, \quad V = \frac{4}{3} \pi r^3.$$

If I have seen farther than others, it is because I have stood on the shoulders of giants.

– Issac Newton

Wallis' identity:

$$\pi = 2 \cdot \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdots}{1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdots}$$

Brouncker's continued fraction expansion:

$$\frac{\pi}{4} = 1 + \frac{1^2}{2 + \frac{3^2}{2 + \frac{5^2}{2 + \frac{7^2}{2 + \cdots}}}}$$

Gregory's series:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots$$

Newton's series:

$$\frac{\pi}{6} = \frac{1}{2} + \frac{1}{2 \cdot 3 \cdot 2^3} + \frac{1 \cdot 3}{2 \cdot 4 \cdot 5 \cdot 2^5} + \cdots$$

Sharp's series:

$$\frac{\pi}{6} = \frac{1}{\sqrt{3}} \left( 1 - \frac{1}{3^1 \cdot 3} + \frac{1}{3^2 \cdot 5} - \frac{1}{3^3 \cdot 7} + \cdots \right)$$

Euler's series:

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \cdots$$

$$\frac{\pi^2}{8} = \frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \frac{1}{9^2} + \cdots$$

$$\frac{\pi^2}{12} = \frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \frac{1}{5^2} - \cdots$$

### Partial Fractions

Let  $N(x)$  and  $D(x)$  be polynomial functions of  $x$ . We can break down  $N(x)/D(x)$  using partial fraction expansion. First, if the degree of  $N$  is greater than or equal to the degree of  $D$ , divide  $N$  by  $D$ , obtaining

$$\frac{N(x)}{D(x)} = Q(x) + \frac{N'(x)}{D(x)},$$

where the degree of  $N'$  is less than that of  $D$ . Second, factor  $D(x)$ . Use the following rules: For a non-repeated factor:

$$\frac{N(x)}{(x-a)D(x)} = \frac{A}{x-a} + \frac{N'(x)}{D(x)},$$

where

$$A = \left[ \frac{N(x)}{D(x)} \right]_{x=a}.$$

For a repeated factor:

$$\frac{N(x)}{(x-a)^m D(x)} = \sum_{k=0}^{m-1} \frac{A_k}{(x-a)^{m-k}} + \frac{N'(x)}{D(x)},$$

where

$$A_k = \frac{1}{k!} \left[ \frac{d^k}{dx^k} \left( \frac{N(x)}{D(x)} \right) \right]_{x=a}.$$

The reasonable man adapts himself to the world; the unreasonable persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable.  
– George Bernard Shaw

Derivatives:

$$1. \frac{d(cu)}{dx} = c \frac{du}{dx}, \quad 2. \frac{d(u+v)}{dx} = \frac{du}{dx} + \frac{dv}{dx}, \quad 3. \frac{d(uv)}{dx} = u \frac{dv}{dx} + v \frac{du}{dx},$$

$$4. \frac{d(u^n)}{dx} = nu^{n-1} \frac{du}{dx}, \quad 5. \frac{d(u/v)}{dx} = \frac{v \left( \frac{du}{dx} \right) - u \left( \frac{dv}{dx} \right)}{v^2}, \quad 6. \frac{d(e^{cu})}{dx} = ce^{cu} \frac{du}{dx},$$

$$7. \frac{d(c^u)}{dx} = (\ln c) c^u \frac{du}{dx}, \quad 8. \frac{d(\ln u)}{dx} = \frac{1}{u} \frac{du}{dx},$$

$$9. \frac{d(\sin u)}{dx} = \cos u \frac{du}{dx}, \quad 10. \frac{d(\cos u)}{dx} = -\sin u \frac{du}{dx},$$

$$11. \frac{d(\tan u)}{dx} = \sec^2 u \frac{du}{dx}, \quad 12. \frac{d(\cot u)}{dx} = -\csc^2 u \frac{du}{dx},$$

$$13. \frac{d(\sec u)}{dx} = \tan u \sec u \frac{du}{dx}, \quad 14. \frac{d(\csc u)}{dx} = -\cot u \csc u \frac{du}{dx},$$

$$15. \frac{d(\arcsin u)}{dx} = \frac{1}{\sqrt{1-u^2}} \frac{du}{dx}, \quad 16. \frac{d(\arccos u)}{dx} = \frac{-1}{\sqrt{1-u^2}} \frac{du}{dx},$$

$$17. \frac{d(\arctan u)}{dx} = \frac{1}{1+u^2} \frac{du}{dx}, \quad 18. \frac{d(\operatorname{arccot} u)}{dx} = \frac{-1}{1+u^2} \frac{du}{dx},$$

$$19. \frac{d(\operatorname{arcsec} u)}{dx} = \frac{1}{u\sqrt{1-u^2}} \frac{du}{dx}, \quad 20. \frac{d(\operatorname{arccsc} u)}{dx} = \frac{-1}{u\sqrt{1-u^2}} \frac{du}{dx},$$

$$21. \frac{d(\sinh u)}{dx} = \cosh u \frac{du}{dx}, \quad 22. \frac{d(\cosh u)}{dx} = \sinh u \frac{du}{dx},$$

$$23. \frac{d(\tanh u)}{dx} = \operatorname{sech}^2 u \frac{du}{dx}, \quad 24. \frac{d(\coth u)}{dx} = -\operatorname{csch}^2 u \frac{du}{dx},$$

$$25. \frac{d(\operatorname{sech} u)}{dx} = -\operatorname{sech} u \tanh u \frac{du}{dx}, \quad 26. \frac{d(\operatorname{csch} u)}{dx} = -\operatorname{csch} u \coth u \frac{du}{dx},$$

$$27. \frac{d(\operatorname{arcsinh} u)}{dx} = \frac{1}{\sqrt{1+u^2}} \frac{du}{dx}, \quad 28. \frac{d(\operatorname{arccosh} u)}{dx} = \frac{1}{\sqrt{u^2-1}} \frac{du}{dx},$$

$$29. \frac{d(\operatorname{arctanh} u)}{dx} = \frac{1}{1-u^2} \frac{du}{dx}, \quad 30. \frac{d(\operatorname{arcoth} u)}{dx} = \frac{1}{u^2-1} \frac{du}{dx},$$

$$31. \frac{d(\operatorname{arcsech} u)}{dx} = \frac{-1}{u\sqrt{1-u^2}} \frac{du}{dx}, \quad 32. \frac{d(\operatorname{arccsch} u)}{dx} = \frac{-1}{|u|\sqrt{1+u^2}} \frac{du}{dx}.$$

Integrals:

$$1. \int cu \, dx = c \int u \, dx, \quad 2. \int (u+v) \, dx = \int u \, dx + \int v \, dx,$$

$$3. \int x^n \, dx = \frac{1}{n+1} x^{n+1}, \quad n \neq -1, \quad 4. \int \frac{1}{x} \, dx = \ln x, \quad 5. \int e^x \, dx = e^x,$$

$$6. \int \frac{dx}{1+x^2} = \arctan x, \quad 7. \int u \frac{dv}{dx} \, dx = uv - \int v \frac{du}{dx} \, dx,$$

$$8. \int \sin x \, dx = -\cos x, \quad 9. \int \cos x \, dx = \sin x,$$

$$10. \int \tan x \, dx = -\ln |\cos x|, \quad 11. \int \cot x \, dx = \ln |\cos x|,$$

$$12. \int \sec x \, dx = \ln |\sec x + \tan x|, \quad 13. \int \csc x \, dx = \ln |\csc x + \cot x|,$$

$$14. \int \arcsin \frac{x}{a} \, dx = \arcsin \frac{x}{a} + \sqrt{a^2 - x^2}, \quad a > 0,$$

15.  $\int \arccos \frac{x}{a} dx = \arccos \frac{x}{a} - \sqrt{a^2 - x^2}, \quad a > 0,$
16.  $\int \arctan \frac{x}{a} dx = x \arctan \frac{x}{a} - \frac{a}{2} \ln(a^2 + x^2), \quad a > 0,$
17.  $\int \sin^2(ax) dx = \frac{1}{2a}(ax - \sin(ax) \cos(ax)),$
18.  $\int \cos^2(ax) dx = \frac{1}{2a}(ax + \sin(ax) \cos(ax)),$
19.  $\int \sec^2 x dx = \tan x,$
20.  $\int \csc^2 x dx = -\cot x,$
21.  $\int \sin^n x dx = -\frac{\sin^{n-1} x \cos x}{n} + \frac{n-1}{n} \int \sin^{n-2} x dx,$
22.  $\int \cos^n x dx = \frac{\cos^{n-1} x \sin x}{n} + \frac{n-1}{n} \int \cos^{n-2} x dx,$
23.  $\int \tan^n x dx = \frac{\tan^{n-1} x}{n-1} - \int \tan^{n-2} x dx, \quad n \neq 1,$
24.  $\int \cot^n x dx = -\frac{\cot^{n-1} x}{n-1} - \int \cot^{n-2} x dx, \quad n \neq 1,$
25.  $\int \sec^n x dx = \frac{\tan x \sec^{n-1} x}{n-1} + \frac{n-2}{n-1} \int \sec^{n-2} x dx, \quad n \neq 1,$
26.  $\int \csc^n x dx = -\frac{\cot x \csc^{n-1} x}{n-1} + \frac{n-2}{n-1} \int \csc^{n-2} x dx, \quad n \neq 1,$
27.  $\int \sinh x dx = \cosh x,$
28.  $\int \cosh x dx = \sinh x,$
29.  $\int \tanh x dx = \ln |\cosh x|,$
30.  $\int \coth x dx = \ln |\sinh x|,$
31.  $\int \operatorname{sech} x dx = \arctan \sinh x,$
32.  $\int \operatorname{csch} x dx = \ln \left| \tanh \frac{x}{2} \right|,$
33.  $\int \sinh^2 x dx = \frac{1}{4} \sinh(2x) - \frac{1}{2} x,$
34.  $\int \cosh^2 x dx = \frac{1}{4} \sinh(2x) + \frac{1}{2} x,$
35.  $\int \operatorname{sech}^2 x dx = \tanh x,$
36.  $\int \operatorname{arcsinh} \frac{x}{a} dx = x \operatorname{arcsinh} \frac{x}{a} - \sqrt{x^2 + a^2}, \quad a > 0,$
37.  $\int \operatorname{arctanh} \frac{x}{a} dx = x \operatorname{arctanh} \frac{x}{a} + \frac{a}{2} \ln |a^2 - x^2|,$
38.  $\int \operatorname{arccosh} \frac{x}{a} dx = \begin{cases} x \operatorname{arccosh} \frac{x}{a} - \sqrt{x^2 + a^2}, & \text{if } \operatorname{arccosh} \frac{x}{a} > 0 \text{ and } a > 0, \\ x \operatorname{arccosh} \frac{x}{a} + \sqrt{x^2 + a^2}, & \text{if } \operatorname{arccosh} \frac{x}{a} < 0 \text{ and } a > 0, \end{cases}$
39.  $\int \frac{dx}{\sqrt{a^2 + x^2}} = \ln \left( x + \sqrt{a^2 + x^2} \right), \quad a > 0,$
40.  $\int \frac{dx}{a^2 + x^2} = \frac{1}{a} \arctan \frac{x}{a}, \quad a > 0,$
41.  $\int \sqrt{a^2 - x^2} dx = \frac{x}{2} \sqrt{a^2 - x^2} + \frac{a^2}{2} \arcsin \frac{x}{a}, \quad a > 0,$
42.  $\int (a^2 - x^2)^{3/2} dx = \frac{x}{8} (5a^2 - 2x^2) \sqrt{a^2 - x^2} + \frac{3a^4}{8} \arcsin \frac{x}{a}, \quad a > 0,$
43.  $\int \frac{dx}{\sqrt{a^2 - x^2}} = \arcsin \frac{x}{a}, \quad a > 0,$
44.  $\int \frac{dx}{a^2 - x^2} = \frac{1}{2a} \ln \left| \frac{a+x}{a-x} \right|,$
45.  $\int \frac{dx}{(a^2 - x^2)^{3/2}} = \frac{x}{a^2 \sqrt{a^2 - x^2}},$
46.  $\int \sqrt{a^2 \pm x^2} dx = \frac{x}{2} \sqrt{a^2 \pm x^2} \pm \frac{a^2}{2} \ln \left| x + \sqrt{a^2 \pm x^2} \right|,$
47.  $\int \frac{dx}{\sqrt{x^2 - a^2}} = \ln \left| x + \sqrt{x^2 - a^2} \right|, \quad a > 0,$
48.  $\int \frac{dx}{ax^2 + bx} = \frac{1}{a} \ln \left| \frac{x}{a + bx} \right|,$
49.  $\int x \sqrt{a + bx} dx = \frac{2(3bx - 2a)(a + bx)^{3/2}}{15b^2},$
50.  $\int \frac{\sqrt{a + bx}}{x} dx = 2\sqrt{a + bx} + a \int \frac{1}{x\sqrt{a + bx}} dx,$
51.  $\int \frac{x}{\sqrt{a + bx}} dx = \frac{1}{\sqrt{2}} \ln \left| \frac{\sqrt{a + bx} - \sqrt{a}}{\sqrt{a + bx} + \sqrt{a}} \right|, \quad a > 0,$
52.  $\int \frac{\sqrt{a^2 - x^2}}{x} dx = \sqrt{a^2 - x^2} - a \ln \left| \frac{a + \sqrt{a^2 - x^2}}{x} \right|,$
53.  $\int x \sqrt{a^2 - x^2} dx = -\frac{1}{3} (a^2 - x^2)^{3/2},$
54.  $\int x^2 \sqrt{a^2 - x^2} dx = \frac{x}{8} (2x^2 - a^2) \sqrt{a^2 - x^2} + \frac{a^4}{8} \arcsin \frac{x}{a}, \quad a > 0,$
55.  $\int \frac{dx}{\sqrt{a^2 - x^2}} = -\frac{1}{a} \ln \left| \frac{a + \sqrt{a^2 - x^2}}{x} \right|,$
56.  $\int \frac{x dx}{\sqrt{a^2 - x^2}} = -\sqrt{a^2 - x^2},$
57.  $\int \frac{x^2 dx}{\sqrt{a^2 - x^2}} = -\frac{x}{2} \sqrt{a^2 - x^2} + \frac{a^2}{2} \arcsin \frac{x}{a}, \quad a > 0,$
58.  $\int \frac{\sqrt{a^2 + x^2}}{x} dx = \sqrt{a^2 + x^2} - a \ln \left| \frac{a + \sqrt{a^2 + x^2}}{x} \right|,$
59.  $\int \frac{\sqrt{x^2 - a^2}}{x} dx = \sqrt{x^2 - a^2} - a \arccos \frac{a}{|x|}, \quad a > 0,$
60.  $\int x \sqrt{x^2 \pm a^2} dx = \frac{1}{3} (x^2 \pm a^2)^{3/2},$
61.  $\int \frac{dx}{x \sqrt{x^2 + a^2}} = \frac{1}{a} \ln \left| \frac{x}{a + \sqrt{a^2 + x^2}} \right|,$

$$\begin{aligned}
62. \int \frac{dx}{x\sqrt{x^2 - a^2}} &= \frac{1}{a} \arccos \frac{a}{|x|}, \quad a > 0, & 63. \int \frac{dx}{x^2\sqrt{x^2 \pm a^2}} &= \mp \frac{\sqrt{x^2 \pm a^2}}{a^2 x}, \\
64. \int \frac{x dx}{\sqrt{x^2 \pm a^2}} &= \sqrt{x^2 \pm a^2}, & 65. \int \frac{\sqrt{x^2 \pm a^2}}{x^4} dx &= \mp \frac{(x^2 + a^2)^{3/2}}{3a^2 x^3}, \\
66. \int \frac{dx}{ax^2 + bx + c} &= \begin{cases} \frac{1}{\sqrt{b^2 - 4ac}} \ln \left| \frac{2ax + b - \sqrt{b^2 - 4ac}}{2ax + b + \sqrt{b^2 - 4ac}} \right|, & \text{if } b^2 > 4ac, \\ \frac{2}{\sqrt{4ac - b^2}} \arctan \frac{2ax + b}{\sqrt{4ac - b^2}}, & \text{if } b^2 < 4ac, \end{cases} \\
67. \int \frac{dx}{\sqrt{ax^2 + bx + c}} &= \begin{cases} \frac{1}{\sqrt{a}} \ln \left| 2ax + b + 2\sqrt{a}\sqrt{ax^2 + bx + c} \right|, & \text{if } a > 0, \\ \frac{1}{\sqrt{-a}} \arcsin \frac{-2ax - b}{\sqrt{b^2 - 4ac}}, & \text{if } a < 0, \end{cases} \\
68. \int \sqrt{ax^2 + bx + c} dx &= \frac{2ax + b}{4a} \sqrt{ax^2 + bx + c} + \frac{4ac - b^2}{8a} \int \frac{dx}{\sqrt{ax^2 + bx + c}}, \\
69. \int \frac{x dx}{\sqrt{ax^2 + bx + c}} &= \frac{\sqrt{ax^2 + bx + c}}{a} - \frac{b}{2a} \int \frac{dx}{\sqrt{ax^2 + bx + c}}, \\
70. \int \frac{dx}{x\sqrt{ax^2 + bx + c}} &= \begin{cases} \frac{-1}{\sqrt{c}} \ln \left| \frac{2\sqrt{c}\sqrt{ax^2 + bx + c} + bx + 2c}{x} \right|, & \text{if } c > 0, \\ \frac{1}{\sqrt{-c}} \arcsin \frac{bx + 2c}{|x|\sqrt{b^2 - 4ac}}, & \text{if } c < 0, \end{cases} \\
71. \int x^3 \sqrt{x^2 + a^2} dx &= \left(\frac{1}{3}x^2 - \frac{2}{15}a^2\right)(x^2 + a^2)^{3/2}, \\
72. \int x^n \sin(ax) dx &= -\frac{1}{a}x^n \cos(ax) + \frac{n}{a} \int x^{n-1} \cos(ax) dx, \\
73. \int x^n \cos(ax) dx &= \frac{1}{a}x^n \sin(ax) - \frac{n}{a} \int x^{n-1} \sin(ax) dx, \\
74. \int x^n e^{ax} dx &= \frac{x^n e^{ax}}{a} - \frac{n}{a} \int x^{n-1} e^{ax} dx, \\
75. \int x^n \ln(ax) dx &= x^{n+1} \left( \frac{\ln(ax)}{n+1} - \frac{1}{(n+1)^2} \right), \\
76. \int x^n (\ln ax)^m dx &= \frac{x^{n+1}}{n+1} (\ln ax)^m - \frac{m}{n+1} \int x^n (\ln ax)^{m-1} dx.
\end{aligned}$$

$$\begin{aligned}
x^1 &= x^1 & x^{\bar{1}} &= x^{\bar{1}} \\
x^2 &= x^2 + x^1 & x^{\bar{2}} &= x^{\bar{2}} - x^{\bar{1}} \\
x^3 &= x^3 + 3x^2 + x^1 & x^{\bar{3}} &= x^{\bar{3}} - 3x^{\bar{2}} + x^{\bar{1}} \\
x^4 &= x^4 + 6x^3 + 7x^2 + x^1 & x^{\bar{4}} &= x^{\bar{4}} - 6x^{\bar{3}} + 7x^{\bar{2}} - x^{\bar{1}} \\
x^5 &= x^5 + 15x^4 + 25x^3 + 10x^2 + x^1 & x^{\bar{5}} &= x^{\bar{5}} - 15x^{\bar{4}} + 25x^{\bar{3}} - 10x^{\bar{2}} + x^{\bar{1}} \\
x^{\bar{1}} &= x^1 & x^1 &= x^1 \\
x^{\bar{2}} &= x^2 + x^1 & x^2 &= x^2 - x^1 \\
x^{\bar{3}} &= x^3 + 3x^2 + 2x^1 & x^3 &= x^3 - 3x^2 + 2x^1 \\
x^{\bar{4}} &= x^4 + 6x^3 + 11x^2 + 6x^1 & x^4 &= x^4 - 6x^3 + 11x^2 - 6x^1 \\
x^{\bar{5}} &= x^5 + 10x^4 + 35x^3 + 50x^2 + 24x^1 & x^{\bar{5}} &= x^5 - 10x^4 + 35x^3 - 50x^2 + 24x^1
\end{aligned}$$

Difference, shift operators:

$$\Delta f(x) = f(x+1) - f(x),$$

$$\mathbb{E} f(x) = f(x+1).$$

Fundamental Theorem:

$$f(x) = \Delta F(x) \Leftrightarrow \sum f(x) \delta x = F(x) + C.$$

$$\sum_a^b f(x) \delta x = \sum_{i=a}^{b-1} f(i).$$

Differences:

$$\Delta(cu) = c\Delta u, \quad \Delta(u+v) = \Delta u + \Delta v,$$

$$\Delta(uv) = u\Delta v + \mathbb{E} v \Delta u,$$

$$\Delta(x^n) = nx^{n-1},$$

$$\Delta(H_x) = x^{-1}, \quad \Delta(2^x) = 2^x,$$

$$\Delta(c^x) = (c-1)c^x, \quad \Delta\binom{x}{m} = \binom{x}{m-1}.$$

Sums:

$$\sum cu \delta x = c \sum u \delta x,$$

$$\sum (u+v) \delta x = \sum u \delta x + \sum v \delta x,$$

$$\sum u \Delta v \delta x = uv - \sum \mathbb{E} v \Delta u \delta x,$$

$$\sum x^n \delta x = \frac{x^{n+1}}{n+1}, \quad \sum x^{-1} \delta x = H_x,$$

$$\sum c^x \delta x = \frac{c^x}{c-1}, \quad \sum \binom{x}{m} \delta x = \binom{x}{m+1}.$$

Falling Factorial Powers:

$$x^{\underline{n}} = x(x-1) \cdots (x-n+1), \quad n > 0,$$

$$x^{\underline{0}} = 1,$$

$$x^{\underline{n}} = \frac{1}{(x+1) \cdots (x+|n|)}, \quad n < 0,$$

$$x^{\underline{n+m}} = x^{\underline{m}}(x-m)^{\underline{n}}.$$

Rising Factorial Powers:

$$x^{\overline{n}} = x(x+1) \cdots (x+n-1), \quad n > 0,$$

$$x^{\overline{0}} = 1,$$

$$x^{\overline{n}} = \frac{1}{(x-1) \cdots (x-|n|)}, \quad n < 0,$$

$$x^{\overline{n+m}} = x^{\overline{m}}(x+m)^{\overline{n}}.$$

Conversion:

$$x^{\underline{n}} = (-1)^n (-x)^{\overline{n}} = (x-n+1)^{\overline{n}}$$

$$= 1/(x+1)^{-\overline{n}},$$

$$x^{\overline{n}} = (-1)^n (-x)^{\underline{n}} = (x+n-1)^{\underline{n}}$$

$$= 1/(x-1)^{-\underline{n}},$$

$$x^n = \sum_{k=1}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\} x^{\underline{k}} = \sum_{k=1}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\} (-1)^{n-k} x^{\overline{k}},$$

$$x^{\underline{n}} = \sum_{k=1}^n \left[ \begin{matrix} n \\ k \end{matrix} \right] (-1)^{n-k} x^k,$$

$$x^{\overline{n}} = \sum_{k=1}^n \left[ \begin{matrix} n \\ k \end{matrix} \right] x^k.$$

Taylor's series:

$$f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \dots = \sum_{i=0}^{\infty} \frac{(x-a)^i}{i!} f^{(i)}(a).$$

Expansions:

$$\begin{aligned} \frac{1}{1-x} &= 1 + x + x^2 + x^3 + x^4 + \dots = \sum_{i=0}^{\infty} x^i, \\ \frac{1}{1-cx} &= 1 + cx + c^2x^2 + c^3x^3 + \dots = \sum_{i=0}^{\infty} c^i x^i, \\ \frac{1}{1-x^n} &= 1 + x^n + x^{2n} + x^{3n} + \dots = \sum_{i=0}^{\infty} x^{ni}, \\ \frac{x}{(1-x)^2} &= x + 2x^2 + 3x^3 + 4x^4 + \dots = \sum_{i=0}^{\infty} ix^i, \\ x^k \frac{d^n}{dx^n} \left( \frac{1}{1-x} \right) &= x + 2^n x^2 + 3^n x^3 + 4^n x^4 + \dots = \sum_{i=0}^{\infty} i^n x^i, \\ e^x &= 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \dots = \sum_{i=0}^{\infty} \frac{x^i}{i!}, \\ \ln(1+x) &= x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{x^i}{i}, \\ \ln \frac{1}{1-x} &= x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \frac{1}{4}x^4 + \dots = \sum_{i=1}^{\infty} \frac{x^i}{i}, \\ \sin x &= x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!}, \\ \cos x &= 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i}}{(2i)!}, \\ \tan^{-1} x &= x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)}, \\ (1+x)^n &= 1 + nx + \frac{n(n-1)}{2}x^2 + \dots = \sum_{i=0}^{\infty} \binom{n}{i} x^i, \\ \frac{1}{(1-x)^{n+1}} &= 1 + (n+1)x + \binom{n+2}{2}x^2 + \dots = \sum_{i=0}^{\infty} \binom{i+n}{i} x^i, \\ \frac{x}{e^x - 1} &= 1 - \frac{1}{2}x + \frac{1}{12}x^2 - \frac{1}{720}x^4 + \dots = \sum_{i=0}^{\infty} \frac{B_i x^i}{i!}, \\ \frac{1}{2x}(1 - \sqrt{1-4x}) &= 1 + x + 2x^2 + 5x^3 + \dots = \sum_{i=0}^{\infty} \frac{1}{i+1} \binom{2i}{i} x^i, \\ \frac{1}{\sqrt{1-4x}} &= 1 + x + 2x^2 + 6x^3 + \dots = \sum_{i=0}^{\infty} \binom{2i}{i} x^i, \\ \frac{1}{\sqrt{1-4x}} \left( \frac{1 - \sqrt{1-4x}}{2x} \right)^n &= 1 + (2+n)x + \binom{4+n}{2}x^2 + \dots = \sum_{i=0}^{\infty} \binom{2i+n}{i} x^i, \\ \frac{1}{1-x} \ln \frac{1}{1-x} &= x + \frac{3}{2}x^2 + \frac{11}{6}x^3 + \frac{25}{12}x^4 + \dots = \sum_{i=1}^{\infty} H_i x^i, \\ \frac{1}{2} \left( \ln \frac{1}{1-x} \right)^2 &= \frac{1}{2}x^2 + \frac{3}{4}x^3 + \frac{11}{24}x^4 + \dots = \sum_{i=2}^{\infty} \frac{H_{i-1} x^i}{i}, \\ \frac{x}{1-x-x^2} &= x + x^2 + 2x^3 + 3x^4 + \dots = \sum_{i=0}^{\infty} F_i x^i, \\ \frac{F_n x}{1 - (F_{n-1} + F_{n+1})x - (-1)^n x^2} &= F_n x + F_{2n} x^2 + F_{3n} x^3 + \dots = \sum_{i=0}^{\infty} F_{ni} x^i. \end{aligned}$$

Ordinary power series:

$$A(x) = \sum_{i=0}^{\infty} a_i x^i.$$

Exponential power series:

$$A(x) = \sum_{i=0}^{\infty} a_i \frac{x^i}{i!}.$$

Dirichlet power series:

$$A(x) = \sum_{i=1}^{\infty} \frac{a_i}{i^x}.$$

Binomial theorem:

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k.$$

Difference of like powers:

$$x^n - y^n = (x-y) \sum_{k=0}^{n-1} x^{n-1-k} y^k.$$

For ordinary power series:

$$\alpha A(x) + \beta B(x) = \sum_{i=0}^{\infty} (\alpha a_i + \beta b_i) x^i,$$

$$x^k A(x) = \sum_{i=k}^{\infty} a_{i-k} x^i,$$

$$\frac{A(x) - \sum_{i=0}^{k-1} a_i x^i}{x^k} = \sum_{i=0}^{\infty} a_{i+k} x^i,$$

$$A(cx) = \sum_{i=0}^{\infty} c^i a_i x^i,$$

$$A'(x) = \sum_{i=0}^{\infty} (i+1) a_{i+1} x^i,$$

$$xA'(x) = \sum_{i=1}^{\infty} i a_i x^i,$$

$$\int A(x) dx = \sum_{i=1}^{\infty} \frac{a_{i-1}}{i} x^i,$$

$$\frac{A(x) + A(-x)}{2} = \sum_{i=0}^{\infty} a_{2i} x^{2i},$$

$$\frac{A(x) - A(-x)}{2} = \sum_{i=0}^{\infty} a_{2i+1} x^{2i+1}.$$

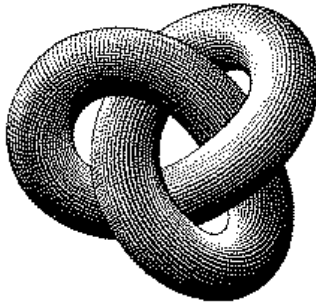
Summation: If  $b_i = \sum_{j=0}^i a_j$  then

$$B(x) = \frac{1}{1-x} A(x).$$

Convolution:

$$A(x)B(x) = \sum_{i=0}^{\infty} \left( \sum_{j=0}^i a_j b_{i-j} \right) x^i.$$

God made the natural numbers;  
all the rest is the work of man.  
– Leopold Kronecker

Expansions:					
$\frac{1}{(1-x)^{n+1}} \ln \frac{1}{1-x}$	$= \sum_{i=0}^{\infty} (H_{n+i} - H_n) \binom{n+i}{i} x^i,$		$\left(\frac{1}{x}\right)^{-n}$	$= \sum_{i=0}^{\infty} \left\{ \begin{matrix} i \\ n \end{matrix} \right\} x^i,$	
$x^{\overline{n}}$	$= \sum_{i=0}^{\infty} \left[ \begin{matrix} n \\ i \end{matrix} \right] x^i,$		$(e^x - 1)^n$	$= \sum_{i=0}^{\infty} \left\{ \begin{matrix} i \\ n \end{matrix} \right\} \frac{n! x^i}{i!},$	
$\left(\ln \frac{1}{1-x}\right)^n$	$= \sum_{i=0}^{\infty} \left[ \begin{matrix} i \\ n \end{matrix} \right] \frac{n! x^i}{i!},$		$x \cot x$	$= \sum_{i=0}^{\infty} \frac{(-4)^i B_{2i} x^{2i}}{(2i)!},$	
$\tan x$	$= \sum_{i=1}^{\infty} (-1)^{i-1} \frac{2^{2i} (2^{2i} - 1) B_{2i} x^{2i-1}}{(2i)!},$	$\zeta(x)$	$= \sum_{i=1}^{\infty} \frac{1}{i^x},$		
$\frac{1}{\zeta(x)}$	$= \sum_{i=1}^{\infty} \frac{\mu(i)}{i^x},$	$\frac{\zeta(x-1)}{\zeta(x)}$	$= \sum_{i=1}^{\infty} \frac{\phi(i)}{i^x},$		
$\zeta(x)$	$= \prod_p \frac{1}{1 - p^{-x}},$				
$\zeta^2(x)$	$= \sum_{i=1}^{\infty} \frac{d(i)}{x^i} \quad \text{where } d(n) = \sum_{d n} 1,$	Stieltjes Integration			
$\zeta(x)\zeta(x-1)$	$= \sum_{i=1}^{\infty} \frac{S(i)}{x^i} \quad \text{where } S(n) = \sum_{d n} d,$	If $G$ is continuous in the interval $[a, b]$ and $F$ is nondecreasing then			
$\zeta(2n)$	$= \frac{2^{2n-1}  B_{2n} }{(2n)!} \pi^{2n}, \quad n \in \mathbb{N},$	$\int_a^b G(x) dF(x)$			
$\frac{x}{\sin x}$	$= \sum_{i=0}^{\infty} (-1)^{i-1} \frac{(4^i - 2) B_{2i} x^{2i}}{(2i)!},$	exists. If $a \leq b \leq c$ then			
$\left(\frac{1 - \sqrt{1-4x}}{2x}\right)^n$	$= \sum_{i=0}^{\infty} \frac{n(2i+n-1)!}{i!(n+i)!} x^i,$	$\int_a^c G(x) dF(x) = \int_a^b G(x) dF(x) + \int_b^c G(x) dF(x).$			
$e^x \sin x$	$= \sum_{i=1}^{\infty} \frac{2^{i/2} \sin \frac{i\pi}{4}}{i!} x^i,$	If the integrals involved exist			
$\sqrt{\frac{1 - \sqrt{1-x}}{x}}$	$= \sum_{i=0}^{\infty} \frac{(4i)!}{16^i \sqrt{2} (2i)! (2i+1)!} x^i,$	$\int_a^b (G(x) + H(x)) dF(x) = \int_a^b G(x) dF(x) + \int_a^b H(x) dF(x),$			
$\left(\frac{\arcsin x}{x}\right)^2$	$= \sum_{i=0}^{\infty} \frac{4^i i!^2}{(i+1)(2i+1)!} x^{2i}.$	$\int_a^b G(x) d(F(x) + H(x)) = \int_a^b G(x) dF(x) + \int_a^b G(x) dH(x),$			
		$\int_a^b c \cdot G(x) dF(x) = \int_a^b G(x) d(c \cdot F(x)) = c \int_a^b G(x) dF(x),$			
		$\int_a^b G(x) dF(x) = G(b)F(b) - G(a)F(a) - \int_a^b F(x) dG(x).$			
		If the integrals involved exist, and $F$ possesses a derivative $F'$ at every point in $[a, b]$ then			
		$\int_a^b G(x) dF(x) = \int_a^b G(x) F'(x) dx.$			
Cramer's Rule				Fibonacci Numbers	
If we have equations:		00 47 18 76 29 93 85 34 61 52 86 11 57 28 70 39 94 45 02 63 95 80 22 67 38 71 49 56 13 04 59 96 81 33 07 48 72 60 24 15 73 69 90 82 44 17 58 01 35 26 68 74 09 91 83 55 27 12 46 30 37 08 75 19 92 84 66 23 50 41 14 25 36 40 51 62 03 77 88 99 21 32 43 54 65 06 10 89 97 78 42 53 64 05 16 20 31 98 79 87		1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... Definitions: $F_i = F_{i-1} + F_{i-2}, \quad F_0 = F_1 = 1,$ $F_{-i} = (-1)^{i-1} F_i,$ $F_i = \frac{1}{\sqrt{5}} \left( \phi^i - \hat{\phi}^i \right),$ Cassini's identity: for $i > 0$ : $F_{i+1}F_{i-1} - F_i^2 = (-1)^i.$ Additive rule: $F_{n+k} = F_k F_{n+1} + F_{k-1} F_n,$ $F_{2n} = F_n F_{n+1} + F_{n-1} F_n.$ Calculation by matrices: $\begin{pmatrix} F_{n-2} & F_{n-1} \\ F_{n-1} & F_n \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n.$	
Let $A = (a_{i,j})$ and $B$ be the column matrix $(b_i)$ . Then there is a unique solution iff $\det A \neq 0$ . Let $A_i$ be $A$ with column $i$ replaced by $B$ . Then					
$x_i = \frac{\det A_i}{\det A}.$					
Improvement makes strait roads, but the crooked roads without Improvement, are roads of Genius. – William Blake (The Marriage of Heaven and Hell)		The Fibonacci number system: Every integer $n$ has a unique representation $n = F_{k_1} + F_{k_2} + \cdots + F_{k_m},$ where $k_i \geq k_{i+1} + 2$ for all $i$ , $1 \leq i < m$ and $k_m \geq 2$ .			







