June 2024-Dec 2024

Samyak Rote

AZ B7 batch notes intern’s all notes

Any problem: contact 8767432663(on whatsapp only)

##### Team: AlgoZenith Alchemist alliance

##### 

### Phase 1

| Title | Notes Link |
| --- | --- |
| C++ Basics | [C++ Basics.pdf](https://drive.google.com/file/d/1JmOshK2NLjpz6VCMfeEXNdoTBcGgGIN2/view?usp=sharing) |
| Is prime checking and divisors |  |
| Gcd rules |  |
| Lcm | a\*b/gcd(a,b) |
| NCR calculations | #include <vector>  using namespace std;  typedef long long ll;  const ll MOD = 1e9 + 7; // Define the modulus  const int MAX = 1e6; // Define the maximum value of n  vector<ll> fact(MAX + 1), inv\_fact(MAX + 1);  void factorials() {  fact[0] = 1;  for (ll i = 1; i <= MAX; i++) {  fact[i] = (fact[i - 1] \* i) % MOD;  }  inv\_fact[MAX] = bin\_pow(fact[MAX], MOD - 2, MOD); // Using Fermat's Little Theorem for modular inversion  for (ll i = MAX - 1; i >= 0; i--) {  inv\_fact[i] = (inv\_fact[i + 1] \* (i + 1)) % MOD;  }  }  // Binary exponentiation  ll bin\_pow(ll base, ll exp, ll mod) {  ll result = 1;  while (exp > 0) {  if (exp % 2 == 1) result = (result \* base) % mod;  base = (base \* base) % mod;  exp /= 2;  }  return result;  }  // nCr calculation using precomputed factorials  ll ncr(int n, int k) {  if (k > n) return 0;  return (fact[n] \* ((inv\_fact[k] \* inv\_fact[n - k]) % MOD)) % MOD;  } |
| 2Pointer | [2 pointers form 2 htx .pdf](https://drive.google.com/file/d/1ckem46CU-T5qUh8rely4jihA3_W8YhZ1/view?usp=sharing)  [2 pointers form 3.pdf](https://drive.google.com/file/d/1CIRu6hu1LcnyPkLFv4cFOpA4JCMv_UAE/view?usp=sharing)[2 pointers.pdf](https://drive.google.com/file/d/1NrpuPky9pb4qZ_wcM97fMbXs-g9eYCaa/view?usp=sharing)[3sum.pdf](https://drive.google.com/file/d/10htCq_CLEuPNJ7pgweISgv03XzAyJV92/view?usp=sharing)[Two pointer variation Spo .pdf](https://drive.google.com/file/d/1kSGOGFyDJd3zN9nZstIKv-culBDE1roO/view?usp=sharing) |
| Binary Search | [Binary search Drill.pdf](https://drive.google.com/file/d/1UJsCPwwV3Szh40dWGfiykg29wij3DONO/view?usp=sharing)[Binary search Foundation.pdf](https://drive.google.com/file/d/1-hswfULXHyaNSUseYvpmuuNcpkyU13Xt/view?usp=sharing)[Binary Seown on Real domain.pdf](https://drive.google.com/file/d/1yp4TYcQB08phsjw5OY30Jd8rue2YkYf_/view?usp=sharing)[BS application 1 ICp .pdf](https://drive.google.com/file/d/1KKh1-NH-34LYQWfT4K8_-PECxNz5_ygX/view?usp=sharing)[BS application 2 Pd1 .pdf](https://drive.google.com/file/d/1hBXQGjW6fbmX-WOYafx9Fxl9u1TKvEZY/view?usp=sharing)[BS application 3 lin .pdf](https://drive.google.com/file/d/1_bzzt4XeSX-7goYRli-8C4PkCx_maQBG/view?usp=sharing)[BS application 4.pdf](https://drive.google.com/file/d/1KJh3Y0wW2piHJQRB0VSL4AoXIE5u8Sdy/view?usp=sharing)[CSEs binary search Te0 .pdf](https://drive.google.com/file/d/1Gjmpo4186zy3vMcXJWwUnT0X3dAPxm95/view?usp=sharing)[Ternary search d01 .pdf](https://drive.google.com/file/d/1xPXm0YaezYNNpjQJ7rBN0rPb-hEtUL1P/view?usp=sharing) |
| Bit Manipulation | [Contribution Techniques in Bits 8qx .pdf](https://drive.google.com/file/d/1w4UHZTYPbwkgyoYTHtG0B3hT_mf6-mIR/view?usp=sharing)[App 4.pdf](https://drive.google.com/file/d/1JiNIkYFdy8GMj6Suh15GXCRoTI5RUfWS/view?usp=sharing)[Bit manipulation 1.pdf](https://drive.google.com/file/d/1dbqRYT4E0h1TR1MOtBIAvWebeMYqdZyz/view?usp=sharing)[Application 3 -Operation decoding.pdf](https://drive.google.com/file/d/1Ueh7UThGwB6WD5hqgD7NU40aPxE8hf46/view?usp=sharing)[cyclic properties of Bits qaz .pdf](https://drive.google.com/file/d/1oZG8BaODDNWrpLxZnlyCnnKoVl2PTGxz/view?usp=sharing)[Bit manipulation 2.pdf](https://drive.google.com/file/d/136YChjyXkdLj-XLCameYL6TT2BQ0Y1_B/view?usp=sharing)[Bitmasking.pdf](https://drive.google.com/file/d/1uk6A5iDiHrBicmLjbMAc48QFac7um13x/view?usp=sharing)[bitset.pdf](https://drive.google.com/file/d/1wcTqfJkTfFQ4OBLC867ks72bd4Mz9Ygw/view?usp=sharing) |
| Dynammic Programming | [22-10-24 DP Drill.pdf](https://drive.google.com/file/d/1csvmn9xV-a-j97u88Y7nHicOxnOS5bhD/view?usp=sharing)[Building strong hold.pdf](https://drive.google.com/file/d/1De6XKdPQJU81Glun1C3xKH7N9bZZR67l/view?usp=sharing)[Day 1.pdf](https://drive.google.com/file/d/1q9CEv7DCPMJa-Wc3mmjUuesQkA1Vhanc/view?usp=sharing)[DP 6-10-24.pdf](https://drive.google.com/file/d/1R23YDilZAkyApj-CoDaElDjePXwY6v1s/view?usp=sharing)[DP 8-10-24.pdf](https://drive.google.com/file/d/1XkRXHu0eSirzFKMWxJmOU_FOEVdmqsEE/view?usp=sharing)[DP 10-10-24.pdf](https://drive.google.com/file/d/1BFPAnJcIj4nkM8mCdbo7_sRq0fLEuLuE/view?usp=sharing)[DP all in one.pdf](https://drive.google.com/file/d/1DrXyUBpmhM_3zVkSWcF7wRk_kU1_s98j/view?usp=sharing)[DP form session.pdf](https://drive.google.com/file/d/1qaoXJ1sQXDK300FRt4Vx3RZAFoAYH9EZ/view?usp=sharing)[Foundation Building.pdf](https://drive.google.com/file/d/13GCgEwv-0QkIX5JwzTXLE2sesflU3KZQ/view?usp=sharing)[Intro to DP.pdf](https://drive.google.com/file/d/1pjTLIaq_oTpJAKHdvEJiCQfoCfpHiTXg/view?usp=sharing)[optimisations.pdf](https://drive.google.com/file/d/1AMvq0co7PFbA54VOYtu5xsuR46v6pZlc/view?usp=sharing) |
| Graphs | [16-9-24.pdf](https://drive.google.com/file/d/1LFTFRAwTjeYIWpjhbiVSeRqMazrSiACq/view?usp=sharing)[Bellman Ford.pdf](https://drive.google.com/file/d/1FX9ZTwEY17sfm-YMjH9U6RiJdtX133cW/view?usp=sharing)[BFS.pdf](https://drive.google.com/file/d/1WSqE5xIt7svpCRWpFxe3Pz6t_B-Ld-fc/view?usp=sharing)[Dijkstra.pdf](https://drive.google.com/file/d/1Ht6nOwJlwvd88VoTj6t6djtzNBj19kKj/view?usp=sharing)[DIS.pdf](https://drive.google.com/file/d/1Yc8Yb7PR33FoC1z9Hp6NIS-xFSWZsOUT/view?usp=sharing)[Floyd Warshall.pdf](https://drive.google.com/file/d/1itzxuSbKhG4z1sEAPMYYB7bujUmHH1Gk/view?usp=sharing)[Graph formulation overview.pdf](https://drive.google.com/file/d/118Bl6aJ2KbwlnkWaFOPX0P_idEk4Uode/view?usp=sharing)[Graph Modelling Ideas.pdf](https://drive.google.com/file/d/1cnvGXqPKHYAr5qo8-MQyVJ5blwnVnN2r/view?usp=sharing)[Introduction.pdf](https://drive.google.com/file/d/1mCg4PFxRpA5KETYNMXgC69Ah44-2OzO0/view?usp=sharing)[MST.pdf](https://drive.google.com/file/d/1W9Ws5aNPJ1lMK6PxvZSNXVfJ2lwkhHOT/view?usp=sharing)[multi source BFS.pdf](https://drive.google.com/file/d/12F5A-W93dSYp0eRHvnWkVxLagqCmQD81/view?usp=sharing)[Graph 18-09-24.pdf](https://drive.google.com/file/d/1wpD3OtdiUIlnhLHZ4Vj5XkreyP9GnJ4C/view?usp=sharing)[ofs tree and cycle detection in directed graph.pdf](https://drive.google.com/file/d/1zd_5w3bp1xc_B6s_8EjfxG5edQKopyOR/view?usp=sharing)[overview.pdf](https://drive.google.com/file/d/1i3f7XSKTqvakhaSgJVdLZD7-HfprslSc/view?usp=sharing)[short notes.pdf](https://drive.google.com/file/d/1FeuVYaxV-yvuZQRhKUhB-YbZTYWiilpl/view?usp=sharing)[shortest path formulation II.pdf](https://drive.google.com/file/d/1jFAmx1LuCCeRKP2-gsbCbbGIqbPDPihU/view?usp=sharing)[shortest path formulation.pdf](https://drive.google.com/file/d/1kVUMMn_mq9oU55Oss2Y3sI6HIaQeJduO/view?usp=sharing)[shortest path formulations.pdf](https://drive.google.com/file/d/1Ikmsht5_9l4EdypeTP03J8V028h2rISq/view?usp=sharing)[Topological ordering.pdf](https://drive.google.com/file/d/132qi2Eq6QfnSr8OOINzBuQlOKFs9hrtF/view?usp=sharing)[Tree Application Idea 1.pdf](https://drive.google.com/file/d/1yRwOQLlDWYtUwW1zjDantFpaWrb0XqYY/view?usp=sharing)[Union find.pdf](https://drive.google.com/file/d/1CvPWEpoOD-42pTPAEzApCeg3lTWTWz0d/view?usp=sharing) |
| Meet In the middle | [4sum.pdf](https://drive.google.com/file/d/15JKY9kdt3bEFC1N5RUDul1u0kUzV8itE/view?usp=sharing)[Meet In Middle foundations.pdf](https://drive.google.com/file/d/1JLCwxV8KRRG3vfnkplxx8frWI922Hikl/view?usp=sharing) |
| Recursion | [Class 1.pdf](https://drive.google.com/file/d/1S2o2_8OCApoX7dzoeaZYUBpCdShyqcP8/view?usp=sharing)[k Knights.pdf](https://drive.google.com/file/d/1VQucuWHYU6iOh7G9CjyirUclRH_S_UXG/view?usp=sharing)[Recursion 2.pdf](https://drive.google.com/file/d/1UEDNWtse33jm9ot-u0QdEOZ6dxxRKKPl/view?usp=sharing) |
| Segment Trees | [27-10-24.pdf](https://drive.google.com/file/d/1JrqeF9y0MhVzfwHYroluZ1Xa3k29qm3X/view?usp=sharing)[Segment Tree Short notes 3zO .pdf](https://drive.google.com/file/d/1N0gDU0TAqKpDUYD_jE66WB0q_PmXTF9s/view?usp=sharing) |
| STL | [5-7-24 STL.pdf](https://drive.google.com/file/d/1b9hXFmEN_wnsn9HWHJJ-xvUlfBGQzveQ/view?usp=sharing)[11,12,13.pdf](https://drive.google.com/file/d/1PfS7mi7zwJPNimrcQQqrWrvCSYd7T0yu/view?usp=sharing)[14,15,16,17.pdf](https://drive.google.com/file/d/1daGO80U0kQIyveX9P9LohgfAKlDJWt38/view?usp=sharing)[Modular Arithmatic and NCR.pdf](https://drive.google.com/file/d/1tECJLuZITGcyGFrwH8bByEO5hmZ7p2wp/view?usp=sharing)[Pattern Matching techniques.pdf](https://drive.google.com/file/d/17yHyEkuQ6hWaJaqzToX43--o3VvcazoA/view?usp=sharing)[problem Solving strategy.pdf](https://drive.google.com/file/d/1NBcZJUpduHMHkpy4WFgz6m7Fr35uOIkv/view?usp=sharing)[STL 3.pdf](https://drive.google.com/file/d/169uETgbjd03scpApYO3cTtrK6J-1IZr4/view?usp=sharing)[STL 6-7-24 (1).pdf](https://drive.google.com/file/d/1iWEScgU48FeMpN6X7fyTKYQ-7E9heBod/view?usp=sharing)[STL app 2 dAi .pdf](https://drive.google.com/file/d/18oo2iS41tBhdQfuBvMi2tpWvB8kq6p53/view?usp=sharing)[stl application 3 (1).pdf](https://drive.google.com/file/d/17j5G0yVmw7V-g8fJX5EtF8UjJi7PQex5/view?usp=sharing)[STL application 4 kfS .pdf](https://drive.google.com/file/d/1d6vwp5CL4_r8pKffZKtNHTmmE9Afdhva/view?usp=sharing)[STL-4.pdf](https://drive.google.com/file/d/1gr1TDkO81dW_zYkqAyzIJjmz1ScEZKsM/view?usp=sharing)[STL-6.pdf](https://drive.google.com/file/d/15N8hctwICozs3KLKHFE707VD4gQZujCm/view?usp=sharing)[stl-application 4.pdf](https://drive.google.com/file/d/1hhnfKrpf_Ldxo97NAJdWFtyJuSzSbCpK/view?usp=sharing) |
| Trie | [5 Nov 2024.pdf](https://drive.google.com/file/d/12m4_taekTiqSdWhw8hwe3b7gnPFXM4bT/view?usp=sharing)[Trie SN wen .pdf](https://drive.google.com/file/d/12YIusGsAMpJU2Ah1fhmRdRlvfHNVMWcj/view?usp=sharing) |
| ShortNotes | [Bit manipulation sn.pdf](https://drive.google.com/file/d/1PYWoNm34pTLHabsUEUv-VWRtw46S4oZb/view?usp=sharing)[short notes.pdf](https://drive.google.com/file/d/1qyFe6GJurcdK6g8ZM3Ll3pVkNuIoQvV0/view?usp=sharing)[short notes.pdf](https://drive.google.com/file/d/1NaQrSyiY_W17lNY7OOZRJK1dZErJehNU/view?usp=sharing) |
| Maths | [BinomialCoef.pdf](https://drive.google.com/file/d/1aZbeAZBG51nL59e8c3XTfxY6FOWqmHRn/view?usp=sharing)[Catalan Number.pdf](https://drive.google.com/file/d/1TydyeAz87sWMmGKro9_H31edlyhPHNNn/view?usp=sharing)[Fibonacci.pdf](https://drive.google.com/file/d/1PvvsukeyNi_7Tuhv6g6ItFIw0SHLYogN/view?usp=sharing)[InclusionExclusion.pdf](https://drive.google.com/file/d/1-M2UrTHW5SNtbZgs_QpEw7JJLttpBgKN/view?usp=sharing)[Josephus.pdf](https://drive.google.com/file/d/12yRrnNnoUwKOF3ly3lWV95ifm9tLCUMf/view?usp=sharing)[Number theory 1.pdf](https://drive.google.com/file/d/14KGZQwE27bTZ10IxewkZCwskv_G1WeBj/view?usp=sharing)[question on maths MeS .pdf](https://drive.google.com/file/d/1HXeVLFDV7x8E1H3iTe7x-uV0F4WuL66V/view?usp=sharing) |
| Prefix Sum | [complete.pdf](https://drive.google.com/file/d/1WW8I_zCSuOjWQ8FmdRc3nmvBdGtz_8Le/view?usp=sharing) |
| SweepLine | [Sweep line.pdf](https://drive.google.com/file/d/1FuGziYl4hA8V2hYI6Cz2W2eElDXIqPeM/view?usp=sharing) |
| Atcoder beautiful solutions | [AtcoderABC397D.pdf](https://drive.google.com/file/d/1FiBwSeeHQuTzlcXVmaoFWNlFDplB7NAE/view?usp=sharing) |
| Leetcode 2045 | class Solution {  public:  vector<vector<int>>g;  vector<int>d1,d2;  vector<bool>v1,v2;  int n;  int ti,ch;  void dijkstra(){  d1.assign(n+1,1e9);  d2.assign(n+1,1e9);  v1.assign(n+1,false);  v2.assign(n+1,false);  auto cmp=[](pair<int,int>a,pair<int,int>b){  return a.first>b.first;  };  priority\_queue<pair<int,int>,vector<pair<int,int>>,decltype(cmp)>pq;  pq.push({0,1});  // v1[1]=true;  // d1[1]=0;  while(!pq.empty()){  int dd=pq.top().first;  int node=pq.top().second;  pq.pop();  if(v2[node])continue;  if(v1[node]){  if(d1[node]==dd)continue;  v2[node]=true;  d2[node]=dd;  int curr=(dd/(2\*ch))\*2\*ch;  int nextTime;  if(dd-curr<ch){  nextTime=dd+ti;  }else{  nextTime=((dd+2\*ch-1)/(2\*ch))\*2\*ch+ti;  }    for(auto x:g[node]){  if(!v1[x]||!v2[x]){  pq.push({nextTime,x});  }  }  }else{  v1[node]=true;  d1[node]=dd;  int curr=(dd/(2\*ch))\*2\*ch;  int nextTime;  if(dd-curr<ch){  nextTime=dd+ti;  }else{  nextTime=((dd+2\*ch-1)/(2\*ch))\*2\*ch+ti;  }    for(auto x:g[node]){  if(!v1[x]||!v2[x]){  pq.push({nextTime,x});  }  }  }  }  }  int secondMinimum(int N, vector<vector<int>>& edges, int time, int change) {  n=N;  ti=time,ch=change;  g.resize(n+1);  for(auto x:edges){  int u=x[0];  int v=x[1];  g[u].push\_back(v);  g[v].push\_back(u);  }  dijkstra();  // cout<<d2[n];  for(int i=1;i<=n;i++){  cout<<d1[i]<<" ";  }  cout<<"\n";  for(int i=1;i<=n;i++){  cout<<d2[i]<<" ";  }  return d2[n];  }  }; |
| DP Form2 session 1 ,  N length Bracket sequence with depth==k | #include<bits/stdc++.h>  using namespace std;  #define int long long  int n,k;  int dp[1010][1010][2];  int rec(int level,int depth,bool seenk){  //prunning  if(depth<0||depth>k)return 0;  //base case  if(level==n){  if(depth==0&&seenk)return 1;  return 0;  }  //cache check  if(dp[level][depth][seenk]!=-1){  return dp[level][depth][seenk];  }  //compute  int ans=rec(level+1,depth+1,seenk||(depth+1)==k);  ans+=rec(level+1,depth-1,seenk||(depth-1)==k);  //save and return  return dp[level][depth][seenk]=ans;  }  void solve(){    cin>>n>>k;  memset(dp,-1,sizeof(dp));  cout<<rec(0,0,0);  }  signed main(){  ios\_base::sync\_with\_stdio(0);  cin.tie(0);  cout.tie(0);  solve();  return 0;  } |
| Generate function | vector<int>cuts;  void generate(int level,int kRem){    //base case  if(level==n){  if(kRem==0)return ;  // return 1e18;  }  //cache checkl  int ans=rec(level,kRem);  //maintain transitions  int mx=-1e18;  int optimalCut;  for(int i=level;i<n;i++){  mx=max(mx,arr[i]);  if(ans==rec(i+1,kRem-1)+mx){  optimalCut=i;  break;  }  }  cuts.push\_back(optimalCut);  generate(optimalCut+1,kRem-1);    } |