jiry

```
#include<cmath>
#include<cstring>
#include<cstdio>
#include<algorithm>
#include<iostream>
using namespace std;
int pd[1<<10],n,m,w[500],A[500],a[500],len,d[500],N,ans,W[500];
unsigned int cover[500],tot,B[500],C[500];
char ch[10];
void findfir(int k1,int k2,int k3,int k4){
	if (k4||(k1==0)){
		unsigned int now=0; int flag=0;
		for (int i=0;i<(1<<n);i++){
			int bo=0;
			for (int j=0;j<n;j++){
				int a=i&(1<<j),b=(k2>>(j*2))&3;
				if ((a==0&&b==2)||(a&&b==1)) bo=1;
			}
			if (bo==0&&pd[i]==0){
				flag=1; break;
			}
			if (bo==0){
				now|=(1ll<<i);
```

```
                }
            }
            if (flag==0){
                len++; A[len]=k2; w[len]=k3; cover[len]=now;
            }
        }
        if (k1==n) return;
        findfir(k1+1,k2,k3,0);
        findfir(k1+1,k2+(1<<(k1*2)),k3+1,1);
        findfir(k1+1,k2+(2<<(k1*2)),k3+1,1);
}
void findans(int k1,int k2,unsigned int k3){
        if (k2>=ans||(k3|B[k1])!=tot) return;
        if (k3==tot){
            ans=k2; return;
        }
        findans(k1+1,k2,k3);
        if ((k3&C[k1])!=C[k1]) findans(k1+1,k2+W[k1],k3|C[k1]);
}
int main(){
        int t; scanf("%d",&t);
        for (;t;t--){
            scanf("%d%d",&n,&m); len=0; ans=n*m;
            memset(pd,0x00,sizeof pd); tot=0;
            for (;m;m--){
```

```
                scanf("%s",ch+1);
                int now=0;
                for (int i=n;i;i--)
                        if (ch[i]>='A'&&ch[i]<='Z') now=(now<<1); else
now=(now<<1)+1;
                pd[now]=1; tot|=(1ll<<now);
        }
        findfir(0,0,0,0);
        memset(d,0x00,sizeof d);
        for (int i=1;i<=len;i++)
                for (int j=1;j<=len;j++)
                        if (i!=j&&(cover[i]&cover[j])==cover[i]&&w[i]>
w[j]) d[i]=1;
        N=0;
        for (int i=1;i<=len;i++) if (d[i]==0) a[++N]=i;
        memset(B,0x00,sizeof B);
        random_shuffle(a+1,a+N+1);
        for (int i=N;i;i--) B[i]=B[i+1]|cover[a[i]];
        for (int i=1;i<=N;i++) C[i]=cover[a[i]],W[i]=w[a[i]];
        findans(1,0,0); printf("%d\n",ans);
    }
    return 0;
}
```

jcvb

```
#include<cstdio>
#include<algorithm>
#include<cassert>
#include<cstring>
#include<iostream>
#include<cstdlib>
#include<cmath>
#include<vector>
#include<map>
#include<set>
#include<queue>
#include<bitset>
using namespace std;
typedef long long ll;
typedef double db;
void gn(int &x){
	int sg=1;char c;while(((c=getchar())<'0'||c>'9')&&c!='-');
	if(c=='-')sg=-1,x=0;else x=c-'0';
	while((c=getchar())>='0'&&c<='9')x=x*10+c-'0';
	x*=sg;
}
void gn(ll &x){
	int sg=1;char c;while(((c=getchar())<'0'||c>'9')&&c!='-');
	if(c=='-')sg=-1,x=0;else x=c-'0';
```

```
	while((c=getchar())>='0'&&c<='9')x=x*10+c-'0';
	x*=sg;
}
int qp(int a,ll b,int mo){int ans=1;do{if(b&1)ans=1ll*ans*a%mo;a=1ll*a*a%mo;}while(b>>=1);return ans;}
int gcd(int a,int b){return b?gcd(b,a%b):a;}
const int mo=1000000007;
int n,m;
int bo[32];
char s[11];

int bb[1111];

int lis[1111];int ltot=0;

int f[1111];
int cnt[1111];
int ne;

int mi;
int cmp(int i,int j){
	return f[i]<f[j];
}
int suf[1111];

void dfs(int i,int cur,int cost){
```

```
        if(cost>=mi)return;
        if(cur==ne){
                mi=min(mi,cost);
                return;
        }
        if((suf[i]|cur)!=ne)return;
        dfs(i+1,cur,cost);
        dfs(i+1,cur|f[lis[i]],cost+cnt[lis[i]]);
}
int main()
{
        int tes;
        gn(tes);
        while(tes--){
                gn(n);gn(m);
                memset(bo,0,sizeof(bo));
                memset(bb,0,sizeof(bb));
                memset(f,0,sizeof(f));
                ltot=0;
                for (int i=1;i<=m;i++){
                        int u=0;
                        scanf("%s",s);
                        for (int j=0;j<n;j++)if(s[j]>='A' && s[j]<='Z')u|=1<<j;
                        //scanf("%d",&u);
                        bo[u]=1;
```

```
}
for (int i=0;i<1<<(2*n);i++){
	int x=i;
	int bo=0;
	int cn=0;
	for (int j=0;j<n;j++){
		if(((x>>(2*j))&3)==3){
			bo=1;
			break;
		}else if(((x>>(2*j))&3)!=0){
			cn++;
		}
	}
	if(bo)continue;
	bb[x]=1;
	cnt[x]=cn;
}
for (int i=0;i<1<<n;i++){
	for (int j=0;j<1<<n;j++){
		int su=0;
		for (int k=0;k<n;k++)if(1<<k&j){
			if(1<<k&i)su|=2<<(k*2);
			else su|=1<<(k*2);
		}
		if(bo[i]==0)bb[su]=0;
```

```
                f[su]|=1<<i;
        }
}
while(1){
        int bo=0;
        for (int i=0;i<1<<(2*n);i++)if(bb[i]){
                for (int j=0;j<n;j++)if((i>>2*j)&3){
                        int x=i^(3<<2*j);
                        if(bb[x]){
                                bb[x]=bb[i]=0;
                                bo=1;
                        }
                }
                for (int j=(i-1)&i;j;j=(j-1)&i){
                        if(bb[j]){
                                bo=1;
                                bb[i]=0;
                                break;
                        }
                }
        }
        if(!bo)break;
}

ne=0;
```

```
            for (int i=0;i<1<<n;i++)if(bo[i])ne|=1<<i;
            mi=1000000000;
            for (int i=0;i<1<<2*n;i++)if(bb[i]){
                  lis[++ltot]=i;
            }
            sort(lis+1,lis+1+ltot,cmp);
            suf[ltot+1]=0;
            for (int i=ltot;i>=1;i--)suf[i]=suf[i+1]|f[lis[i]];
            dfs(1,0,0);
            printf("%d\n",mi);
      }
      return 0;
}
```

matthew

```
using namespace std;

#define REP(i, a, b) for (int i = (a), _end_ = (b); i < _end_; ++i)
#define debug(...) fprintf(stderr, __VA_ARGS__)
#define mp make_pair
#define x first
#define y second
#define pb push_back
```

```
#define SZ(x) (int((x).size()))
#define ALL(x) (x).begin(), (x).end()

#ifdef __WIN32
#define Rand() (rand() << 15 | rand())
#define LLFORMAT "I64"
#else
#define Rand() rand()
#define LLFORMAT "11"
#endif

template<typename T> inline bool chkmin(T &a, const T &b) { return a > b ? a = b, 1 : 0; }
template<typename T> inline bool chkmax(T &a, const T &b) { return a < b ? a = b, 1 : 0; }

typedef long long LL;

const int oo = 0x3f3f3f3f;

const int max0 = 243, max1 = 32;

int has;

vector<pair<int, int> > all;
```

```
int val0[max0 + 5], val1[max0 + 5];


int ans = oo;


int suf[max0 + 5];


int tot = 0;


int dp[max1 + 5];


int cnt = 0;


void dfs(const int &x, const int &y)
{
	++cnt;
	if (cnt >= 1.2e6) return;
	if (tot + dp[__builtin_popcount(x ^ has)] >= ans) return;
	if (x == has)
	{
		chkmin(ans, tot);
		return;
	}
	if ((x | suf[y]) != has) return;
	if ((x | val0[y]) != x) tot += val1[y], dfs(x | val0[y], y + 1),
tot -= val1[y];
	dfs(x, y + 1);
```

```
}

inline bool cmp(const pair<int, int> &x, const pair<int, int> &y)
{
	return x.y < y.y;
}

int greedy(int x)
{
	if (x == has) return 0;
	double Max = -1;
	int Maxval = -1;
	int Maxadd = -1;
	for (auto it : all)
	{
		double delta = double(__builtin_popcount((x | it.x) ^ x)) /
it.y;
		if (chkmax(Max, delta)) Maxval = x | it.x, Maxadd = it.y;
	}
	return greedy(Maxval) + Maxadd;
}

int main()
{
	srand(time(NULL));
#ifndef ONLINE_JUDGE
```

```
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif
	int T;
	scanf("%d", &T);
	cnt = 0;
	while (T--)
	{
		has = 0;
		int n, m;
		scanf("%d%d", &n, &m);
		REP(i, 0, m)
		{
			static char s[100];
			scanf("%s", s);
			int tmp = 0;
			REP(j, 0, n) (tmp <<= 1) |= s[j] >= 'a';
			has |= 1 << tmp;
		}
		all.clear();
		REP(i, 0, 1 << n)
			for (int j = i; ; j = (j - 1) & i)
			{
				int tmp = 0;
				bool flag = 1;
```

```
                        REP(k, 0, 1 << n)
                                if ((k & i) == j)
                                {
                                        if (!(has >> k & 1))
                                        {
                                                flag = 0;
                                                break;
                                        }
                                        tmp |= (1 << k);
                                }
                        if (flag) all.pb(mp(tmp, __builtin_popcount(i)));
                        if (!j) break;
                }
        vector<pair<int, int> > newall;
        REP(i, 0, SZ(all))
        {
                bool flag = 1;
                REP(j, 0, SZ(all))
                        if (i != j && all[j].y <= all[i].y && ((all[i].x
& all[j].x) == all[i].x) && (all[i].x != all[j].x || all[i].y != all[j].y
|| i > j)) flag = 0;
                if (flag) newall.pb(all[i]);
        }
        all = newall;
        random_shuffle(ALL(all));
        stable_sort(ALL(all), cmp);
```

```
            memset(dp, oo, sizeof dp);
            dp[0] = 0;
            REP(i, 0, SZ(all))
            {
                  auto it = all[i];
                  int lyc = __builtin_popcount(it.x);
                  for (int j = max1 - lyc; j >= 0; --j)
                        chkmin(dp[j + lyc], dp[j] + it.y);
            }
            for (int j = max1 - 1; j >= 0; --j) chkmin(dp[j], dp[j + 1]);
            suf[SZ(all)] = 0;
            for (int i = SZ(all) - 1; i >= 0; --i) suf[i] = suf[i + 1] |
all[i].x;
            REP(i, 0, SZ(all)) val0[i] = all[i].x, val1[i] = all[i].y;
            ans = greedy(0);
            tot = 0;
            cnt = 0;
            dfs(0, 0);
            printf("%d\n", ans);
      }
//    debug("%ld\n", clock());
      return 0;
}
```

weng

```
#include<cstdio>
#include<cstring>
#include<algorithm>

using namespace std;

const int MAXN = 1005;

typedef long long LL;

char s[7];
LL Cover[MAXN],Len[MAXN],Bak[MAXN],Obj;
bool Apear[100],None[MAXN];
int V[MAXN],First[MAXN],Fi[MAXN][5],Fac[6],N,M,Ans,cnt;

bool Upper(char c)
{
	return (c >= 'A' && c <= 'Z');
}

bool Chk(int Tag,int Use)
{
	int ans = 0;
	for(int i = 1;i <= M;i ++)
		if ((V[i] ^ Tag) == 0) ans ++;
```

```
	if (ans == Fac[N - Use]) return 1;
	return 0;
}

void Dfs(int Now,int Tag,int Use,int Cur)
{
	if (Now >= N && Chk(Tag,Use))
	{
		Len[Cur] = Use;
		Ans += Use;
		for(int i = 1;i <= M;i ++)
			if ((V[i] ^ Tag) == 0) Cover[Cur] |= (111 << (i - 1));
	}
	if (Now >= N) return;
	Dfs(Now + 1,Tag,Use,Cur * 3);//Not Use this member
	for(int j = 1;j <= M;j ++)
		V[j] |= (Fi[j][Now] * Fac[Now]);
	Dfs(Now + 1,Tag | Fac[Now],Use + 1,Cur * 3 + 1);
	Dfs(Now + 1,Tag,Use + 1,Cur * 3 + 2);
	for(int j = 1;j <= M;j ++)
		V[j] -= (Fi[j][Now] * Fac[Now]);
}

void Dfs(int Now,LL Cur,int Use)
{
```

```
	if (Use >= Ans) return;
	if (Cur == Obj) {Ans = Use;return;}
	if (Now == cnt) return;
	if ((Cur | Bak[Now]) != Obj) return;
	if ((Cur | Cover[Now]) != Cur) Dfs(Now + 1,Cur | Cover[Now],Use +
Len[Now]);
	Dfs(Now + 1,Cur,Use);
}

void Work()
{
	memset(Apear,0,sizeof Apear),memset(First,0,sizeof
First);memset(Cover,0,sizeof Cover),memset(Bak,0,sizeof Bak);
	memset(None,0,sizeof None);
	Ans = 0;
	scanf("%d%d", &N, &M);
	for(int i = 1;i <= M;i ++)
	{
		scanf("%s", s);
		for(int j = 0;j < N;j ++)
		{
			Fi[i][j] = Upper(s[j]);
			First[i] |= (Fi[i][j] * Fac[j]);
		}
	}
	for(int i = 1;i <= M;i ++)
```

```
            if (Apear[First[i]])
swap(First[i],First[M]),swap(Fi[i],Fi[M]),M --,i --; else
                  Apear[First[i]] = 1;
      Dfs(0,0,0,0);
      cnt = 0;
      for(int i = 0;i <= 243;i ++)
            if (Cover[i]) Cover[cnt ++] = Cover[i],Len[cnt - 1] = Len[i];
      for(int i = 0;i < cnt;i ++)
            if (!None[i])
                  for(int j = 0;j < cnt;j ++)
                        if (i != j && (Cover[i] & Cover[j]) == Cover[j]
&& Len[i] <= Len[j])
                              None[j] = 1;
      int tot = 0;
      for(int i = 0;i < cnt;i ++)
            if (!None[i])
                  Cover[tot ++] = Cover[i],Len[tot - 1] = Len[i];
      cnt = tot;
      for(int i = cnt - 1;i + 1;i --)
            Bak[i] = (Bak[i + 1] | Cover[i]);
      Obj = (1 << M) - 1;
      Ans = (1 << 30);
      Dfs(0,0,0);
      printf("%d\n", Ans);
}
```

```
int main()
{
	for(int j = 0;j <= 5;j ++) Fac[j] = (1 << j);
	int T;
	scanf("%d", &T);
	for(;T;T --) Work();
	return 0;
}
```

stillwell

```
using namespace std;

int Case,T,n,m,i,j,k,l,ans;
int c[1000],d[1000],h[1000],w[1000],tot,cnt;
unsigned int bit[1000],e[1000],BIT,aim;
int son[1000],next[10000],ed[10000],ltot;
bool f[2005],g[2005],del[1000];
char s[10];

map <unsigned int,int> Hash[10],M[1000][235];

void dfs(int x,int y,int z)
```

```
{
	if(x>n)
	{
		int sum=0;
		for(int i=1;i<=tot;++i)if((c[i]&y)==y)++sum;
		if(sum==(1<<(n-z)))d[++cnt]=y;
		if(z==n)BIT=(BIT<<1)+f[y];
		return;
	}
	dfs(x+1,y*4+1,z+1);
	dfs(x+1,y*4+2,z+1);
	dfs(x+1,y*4,z);
}

int calc(int k)
{
	int sum=0;
	for(int i=1;i<=n;++i)
	{
		if(k%4)++sum;
		k/=4;
	}
	return sum;
}
```

```
void work(int x,int y,int z,unsigned int B)
{
	if(y>=ans)return;
	int i;
	if(B==aim)
	{
		ans=y;
		return;
	}
	if(x>cnt)return;
	z=B%233;
	if(M[x][z].find(B)==M[x][z].end())M[x][z][B]=y;
	else
	{
		if(y>=M[x][z][B])return;
		M[x][z][B]=y;
	}
	int k;
	if((e[x]&B)!=e[x])
	{
		k=0;
		for(i=e[x]^(e[x]&B);i;i-=i&-i)++k;
		work(x+1,y+h[x],z+k,B|e[x]);
	}
	k=1;
```

```
        for(i=son[x];i;i=next[i])
        {
                --w[ed[i]];
                if(!w[ed[i]])k=0;
        }
        if(k)work(x+1,y,z,B);
        for(i=son[x];i;i=next[i])
        {
                ++w[ed[i]];
        }
}

int main()
{
        scanf("%d",&T);
        for(Case=1;Case<=T;++Case)
        {
                scanf("%d%d",&n,&m);
                memset(f,false,sizeof(f));
                memset(g,false,sizeof(g));
                tot=cnt=0;
                for(;m;--m)
                {
                        scanf("%s",s+1);
                        k=0;
```

```
        for(i=1;i<=n;++i)
        if(s[i]>='a')k=k*4+1;
        else k=k*4+2;
        if(!f[k])c[++tot]=k;
        f[k]=true;
}
BIT=0;
dfs(1,0,0);
if(Hash[n].find(BIT)!=Hash[n].end())
{
        printf("%d\n",Hash[n][BIT]);
        continue;
}
for(i=1;i<=cnt;++i)
{
        k=d[i];
        for(j=1;j<=cnt;++j)
        if(d[j]!=k&&((k&d[j])==k))
        {
                d[j]=d[cnt];
                --cnt;--j;
        }
}
sort(d+1,d+cnt+1);
for(i=1;i<=cnt;++i)del[i]=false;
```

```
		for(i=1;i<=cnt;++i)h[i]=calc(d[i]);
		for(i=1;i<=cnt;++i)
		{
			bit[i]=0;
			for(j=1;j<=tot;++j)bit[i]=(bit[i]
<<1)+((d[i]&c[j])==d[i]);
		}
		for(i=1;i<=cnt;++i)
		for(j=i+1;j<=cnt;++j)
		{
			if((bit[i]&bit[j])==bit[j])
			del[j]=true;
		}
		j=0;
		for(i=1;i<=cnt;++i)
		if(!del[i])
		{
			++j;
			if(i!=j)d[j]=d[i];
		}
		cnt=j;
		for(i=1;i<=cnt;++i)h[i]=calc(d[i]);
		for(i=1;i<=tot;++i)w[i]=0;
		for(i=1;i<=cnt;++i)
		for(j=1;j<=tot;++j)
		if((d[i]&c[j])==d[i])
```

```
            ++w[j];
            ans=n*tot;
            for(i=1;i<=cnt;++i)son[i]=0;
            ltot=0;
            for(i=1;i<=cnt;++i)e[i]=0;
            for(i=1;i<=cnt;++i)
            for(j=1;j<=tot;++j)
            if((d[i]&c[j])==d[i])
            {
                ++ltot;next[ltot]=son[i];son[i]=ltot;ed[ltot]=j;
                e[i]|=((unsigned int)1)<<j-1;
            }
            for(i=1;i<=cnt;++i)
            for(j=0;j<=233;++j)
            M[i][j].clear();
            aim=0;
            for(i=1;i<=tot;++i)aim=aim<<1|1;
            work(1,0,0,0);
            printf("%d\n",ans);
            Hash[n][BIT]=ans;
        }
}
```

wwt

```
#include <cstdio>
#include <algorithm>
#include <map>
using namespace std;

const int N=5,M=1005,S=250,inf=(int)1e9+7;
const unsigned P=701;

int n,m,a[M],pw3[N+1],len[S],st[S],cov[M];
bool sub[S][S],cho[S];
unsigned aim;
map<unsigned,int> memo[S][P];

int search(int i=0,unsigned bit=0U){
	if(bit==aim)
		return 0;
	++i;
	if(memo[i][bit%P].find(bit)!=memo[i][bit%P].end())
		return memo[i][bit%P][bit];
	int &ans=memo[i][bit%P][bit];
	ans=inf;
	cho[i]=true;
	unsigned bit_=bit;
	for(int j=1;j<=m;j++) if(sub[a[j]][st[i]]) bit_|=1U<<(j-1);
	if(bit_!=bit)
```

```
            ans=min(ans,len[st[i]]+search(i,bit_));
        cho[i]=false;
        bool fine=true;
        for(int j=1;j<=m;j++) if(sub[a[j]][st[i]]) if(!--cov[j]) fine=false;
        if(fine)
            ans=min(ans,search(i,bit));
        for(int j=1;j<=m;j++) if(sub[a[j]][st[i]]) cov[j]++;
        return ans;
}

int Main(){
        scanf("%d%d",&n,&m);
        pw3[0]=1;
        for(int i=1;i<=n;i++) pw3[i]=pw3[i-1]*3;
        for(int s=0;s<pw3[n];s++)
            for(int t=0;t<pw3[n];t++){
                bool ok=true;
                for(int j=0;j<n;j++){
                    int x=s/pw3[j]%3,y=t/pw3[j]%3;
                    if(x){
                        if(y&&x!=y) ok=false;
                    }
                    else{
                        if(y) ok=false;
                    }
```

```
                if(!ok) break;
            }
            sub[s][t]=ok;
        }
    for(int i=1;i<=m;i++){
        static char s[100];
        scanf("%s",s);
        a[i]=0;
        for(int j=0;j<n;j++) a[i]+=(s[j]>='a'?2:1)*pw3[j];
    }
    sort(a+1,a+m+1);
    m=unique(a+1,a+m+1)-(a+1);
    aim=0;
    for(int i=0;i<m;i++) aim=aim<<1|1U;
    *st=0;
    for(int s=0;s<pw3[n];s++){
        bool ok=true;
        for(int i=1;i<=*st;i++) if(sub[s][st[i]]){
            ok=false; break;
        }
        if(!ok) continue;
        len[s]=0;
        for(int j=0;j<n;j++) if(s/pw3[j]%3) len[s]++;
        int cnt=0;
        for(int i=1;i<=m;i++) if(sub[a[i]][s]) cnt++;
```

```
            if(cnt==(1<<(n-len[s])))
                st[++*st]=s;
        }
        for(int j=1;j<=m;j++){
            cov[j]=0;
            for(int i=1;i<=*st;i++)
                if(sub[a[j]][st[i]]) cov[j]++;
        }
        for(int i=1;i<=*st;i++)
            for(int r=0;r<P;r++) memo[i][r].clear();
        return search();
}

int main(){
        int T; scanf("%d",&T);
        while(T--) printf("%d\n",Main());
}
```