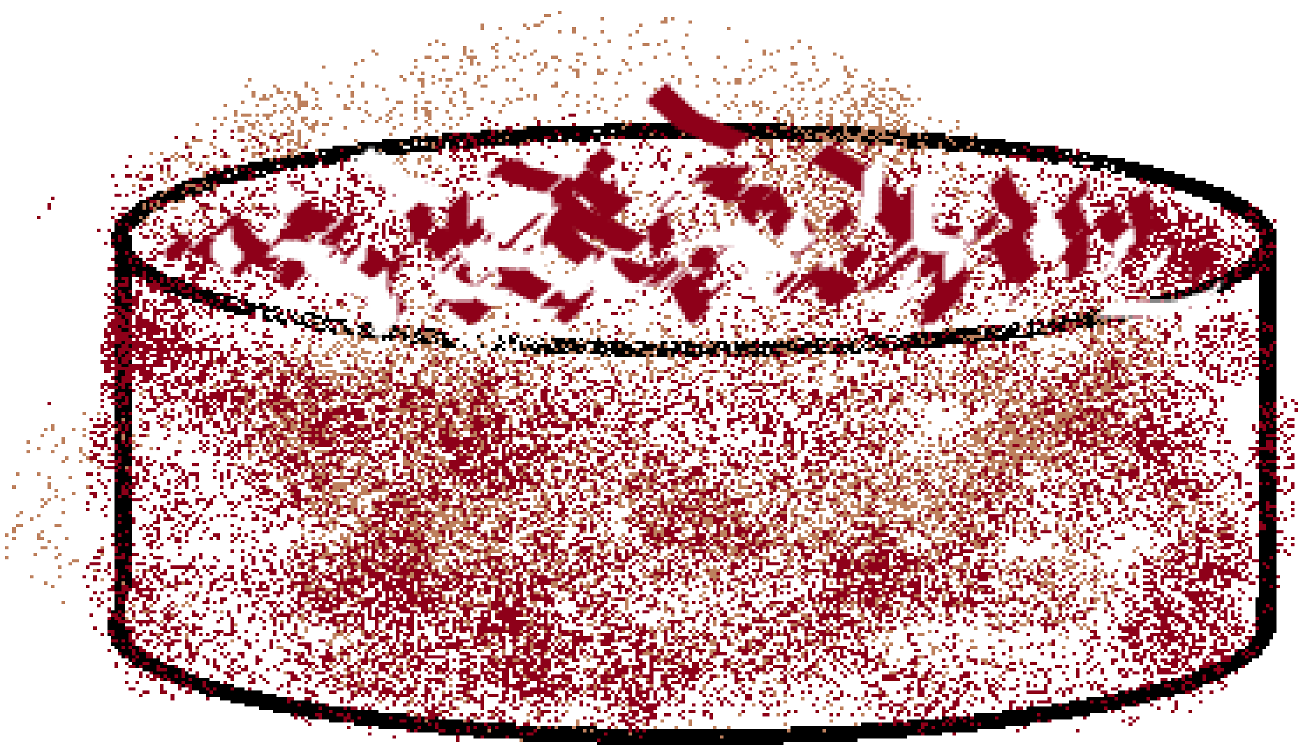
Cake 2　解説



解説：　秀 郁未 (tozangezan)

"2" …?

ケーキの切り分け (Cake)

JOI くんと IOI ちゃんは双子の兄妹である．JOI くんは最近お菓子作りに凝っていて，今日も JOI くんはケーキを焼いて食べようとしたのだが，焼きあがったところで匂いをかぎつけた IOI ちゃんが来たので 2 人でケーキを分けることになった．

ケーキは円形である．ある点から放射状に切り目を入れ，ケーキを N 個のピースに切り分け，ピースに 1 から N まで反時計回りに番号をつけた．つまり，$1 \leqq i \leqq N$ に対し，i 番目のピースは $i-1$ 番目と $i+1$ 番目のピースと隣接している (ただし 0 番目は N 番目，$N+1$ 番目は 1 番目とみなす)．i 番目のピースの大きさは A_i だったが，切り方がとても下手だったので A_i はすべて異なる値になった．

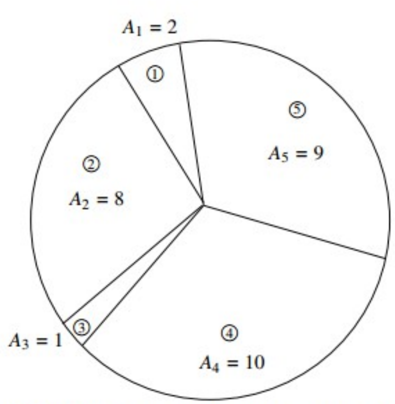


図 1: ケーキの例 ($N = 5, A_1 = 2, A_2 = 8, A_3 = 1, A_4 = 10, A_5 = 9$)

2

ケーキの切り分け 2 (Cake 2)

JOI くんと IOI ちゃんは双子の兄妹である．JOI くんは最近お菓子作りに凝っていて，今日も JOI くんはケーキを焼いて食べようとしたのだが，焼きあがったところで匂いをかぎつけた IOI ちゃんが来たので 2 人でケーキを分けることになった．

ケーキは円形である．ある点から放射状に切り目を入れ，ケーキを N 個のピースに切り分け，ピースに 1 から N まで反時計回りに番号をつけた．つまり，$1 \leqq i \leqq N$ に対し，i 番目のピースは $i-1$ 番目と $i+1$ 番目のピースと隣接している (ただし 0 番目は N 番目，$N+1$ 番目は 1 番目とみなす)．i 番目のピースの大きさは A_i だったが，切り方がとても下手だったので A_i はすべて異なる値になった．

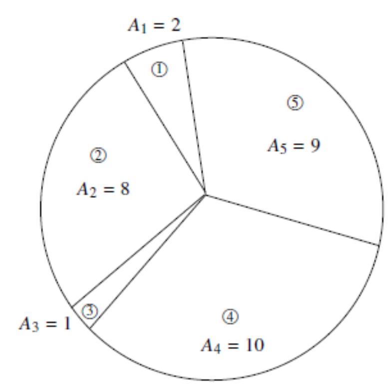
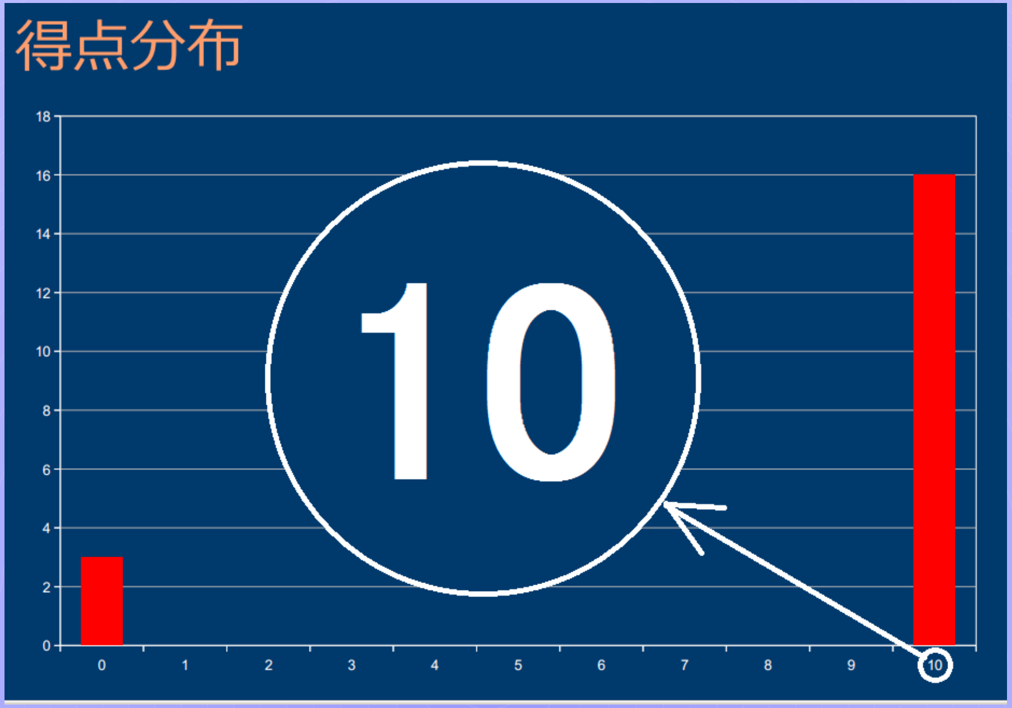


図 1: ケーキの例 ($N = 5, A_1 = 2, A_2 = 8, A_3 = 1, A_4 = 10, A_5 = 9$)

冒頭、完全に一致

- Cake(1): 2013春合宿 Day3
- 得点分布(満点は100です)



で、この問題は…

- 中身は全然違います
- そもそも本選2なのでこんなに難しいはずない
- 見たことある設定だからといって変な読み飛ばしとか誤読とかをしてはいけない

概要

- 環状で数が並んでいる
- JOIくん、IOIちゃんが次のように数を取っていく
  - JOIくんが最初にすきなところを取る
  - IOIちゃんは両端のうち大きいほうを取る
  - JOIくんは両端のうちすきなほうを取る
  - IOIちゃんは両端のうち大きいほうを取る

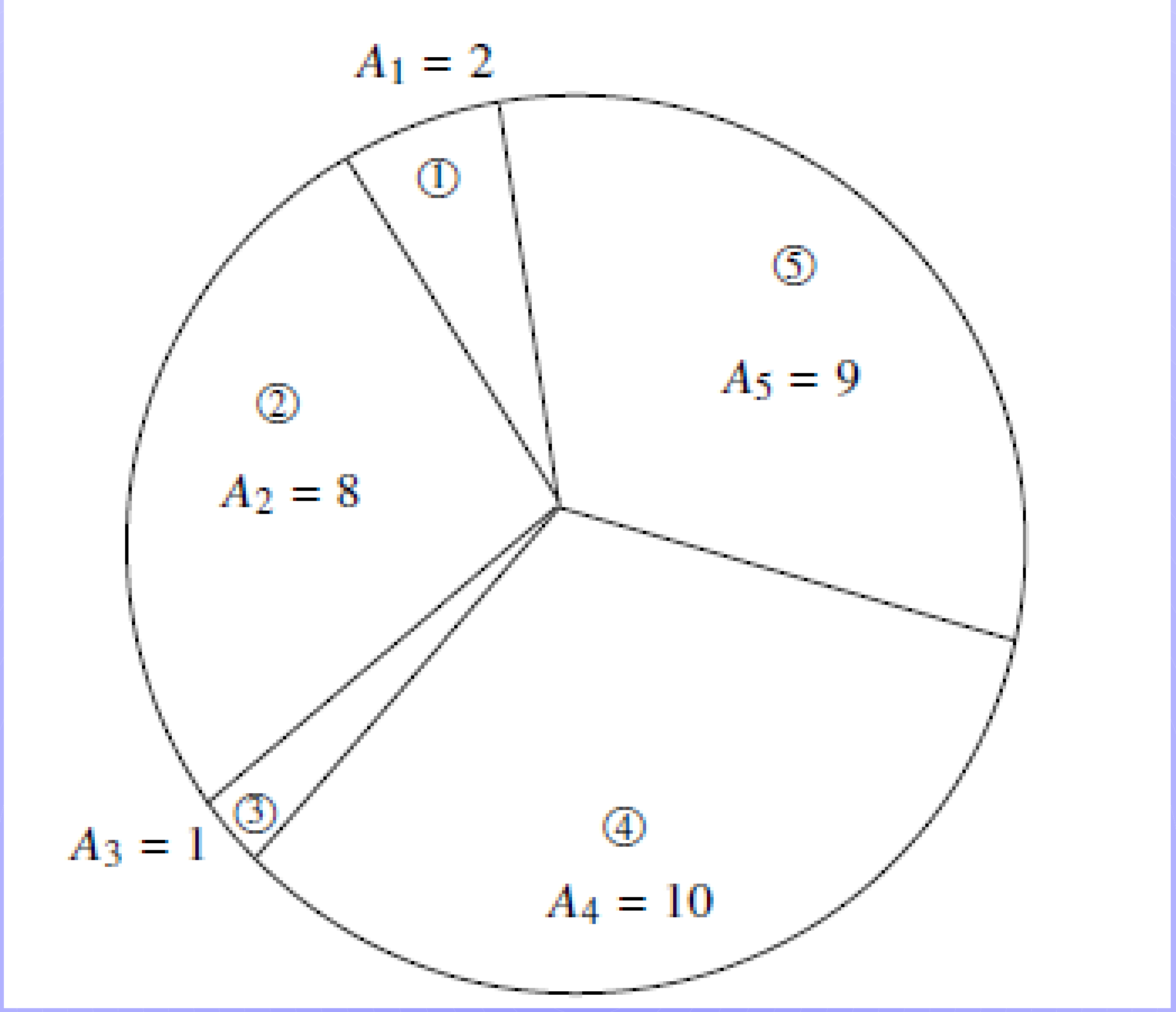
概要

- JOIくんが最適に操作したときに、JOIくんが取れる数の合計の最大値は？
- N ≦ 2,000
- IOIちゃんには戦略はありません(残ったものの両端のうち大きいほうをとるだけ)

例

JOI KUN

0



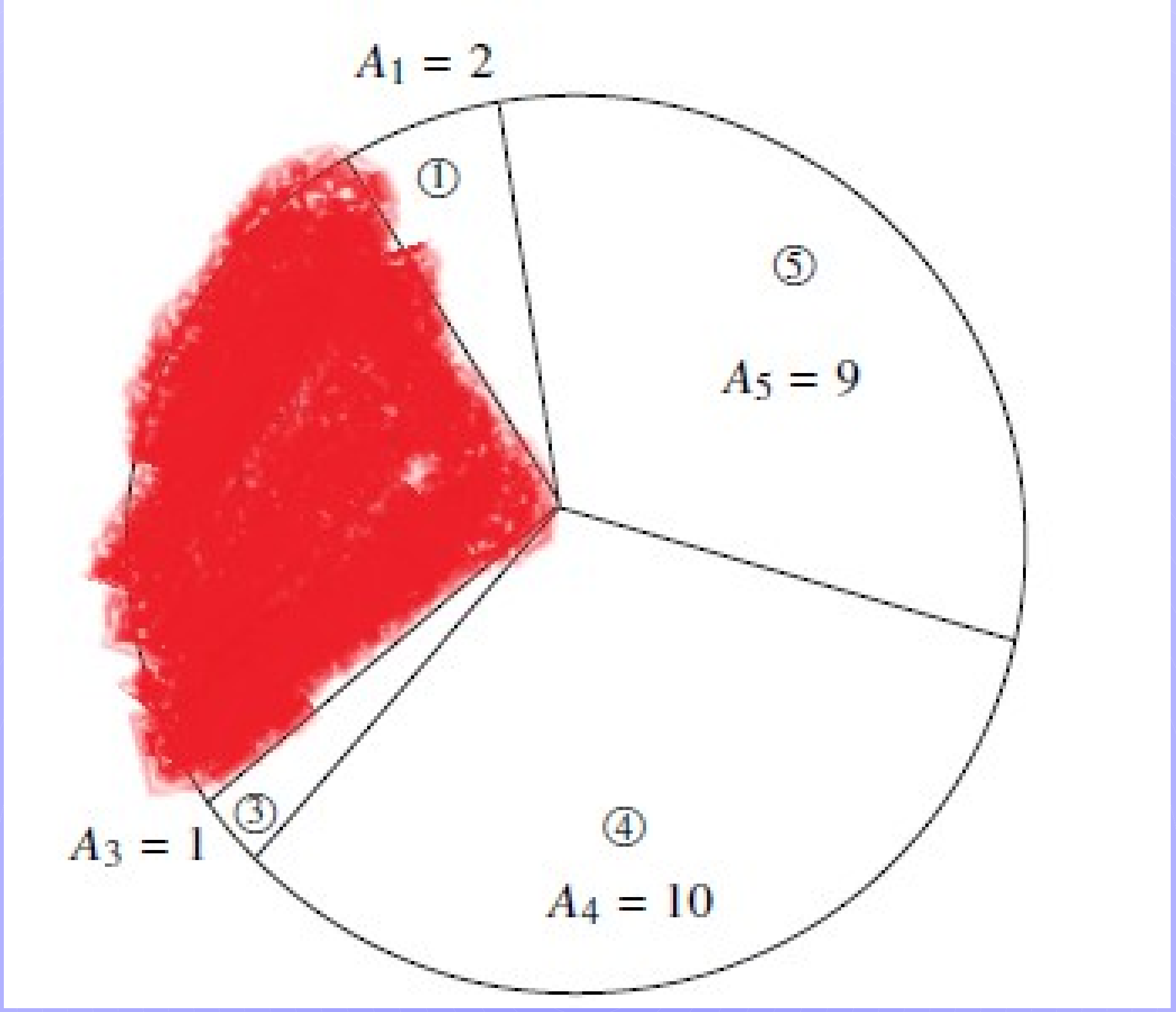
IOI CHAN

0

例

JOI KUN

8



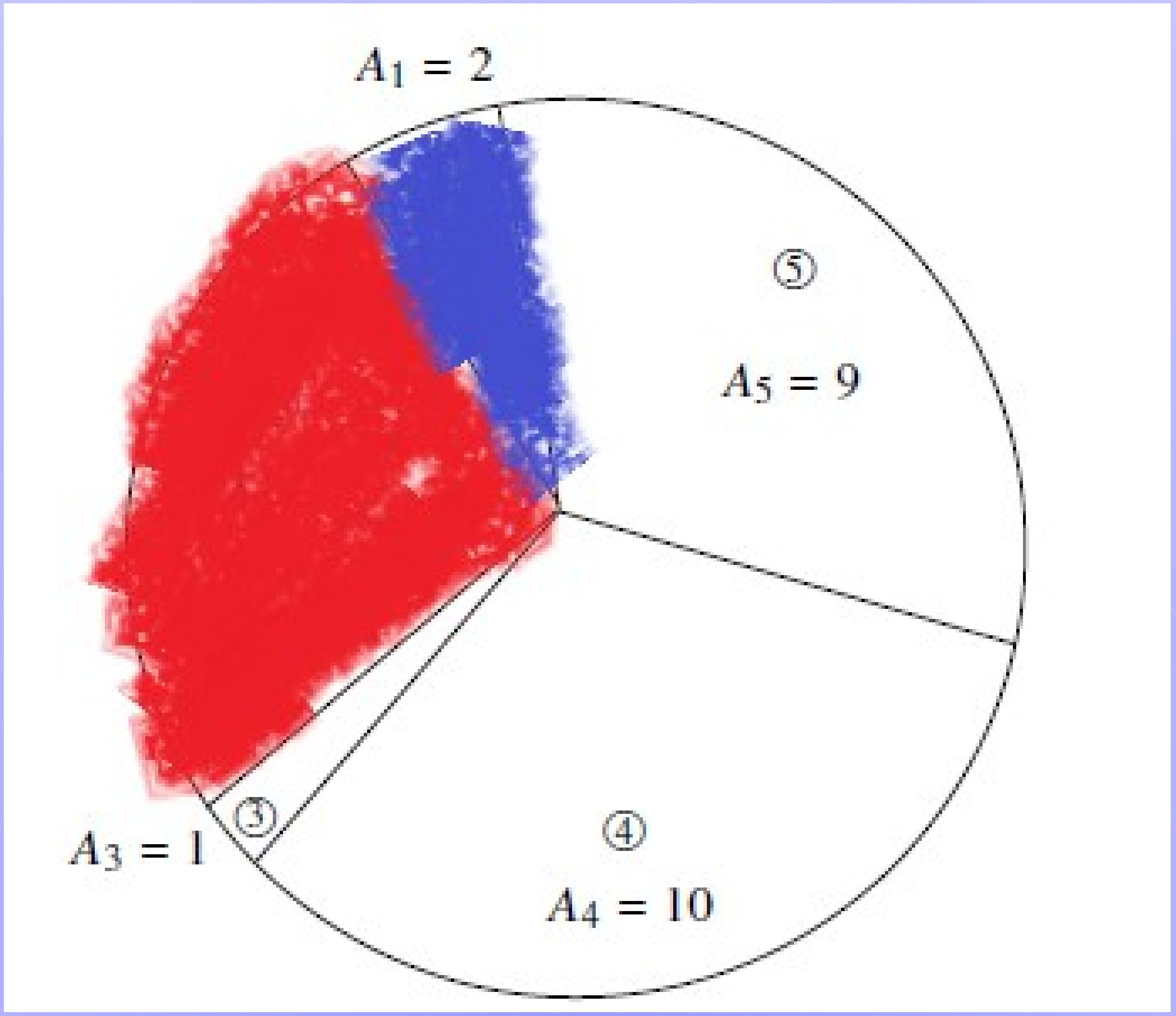
IOI CHAN

0

例

JOI KUN

8



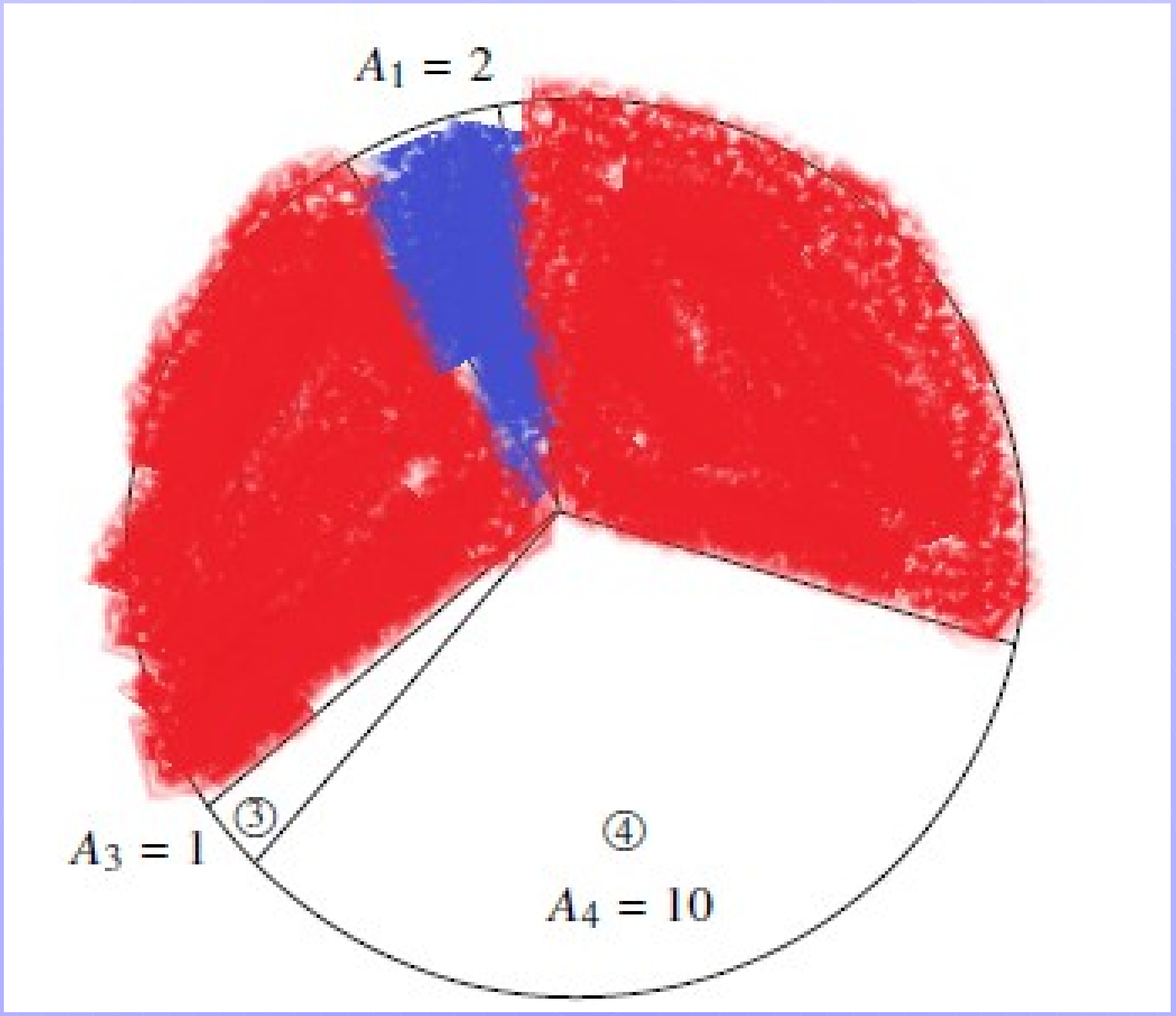
IOI CHAN

2

例

JOI KUN

17



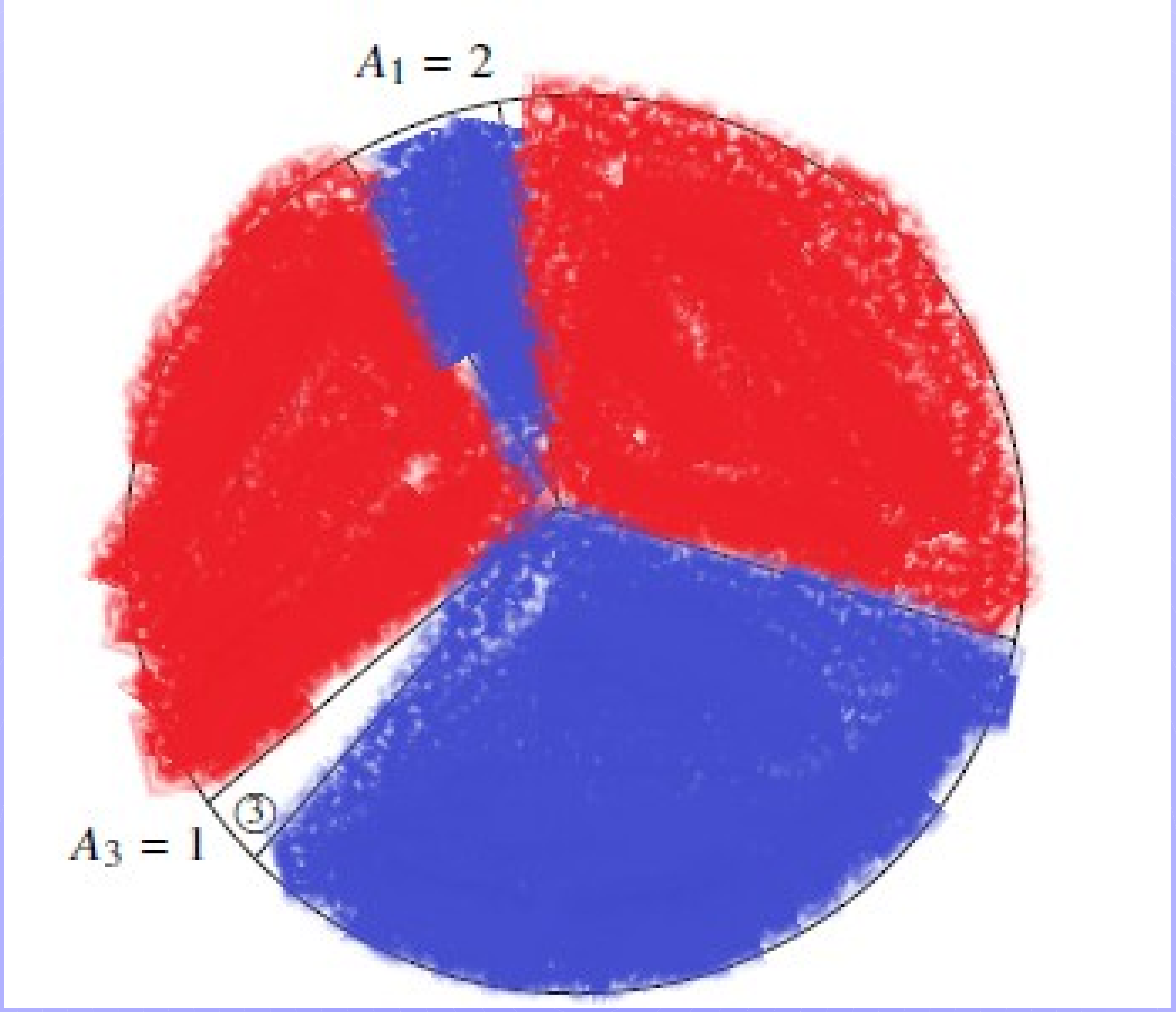
IOI CHAN

2

例

JOI KUN

17



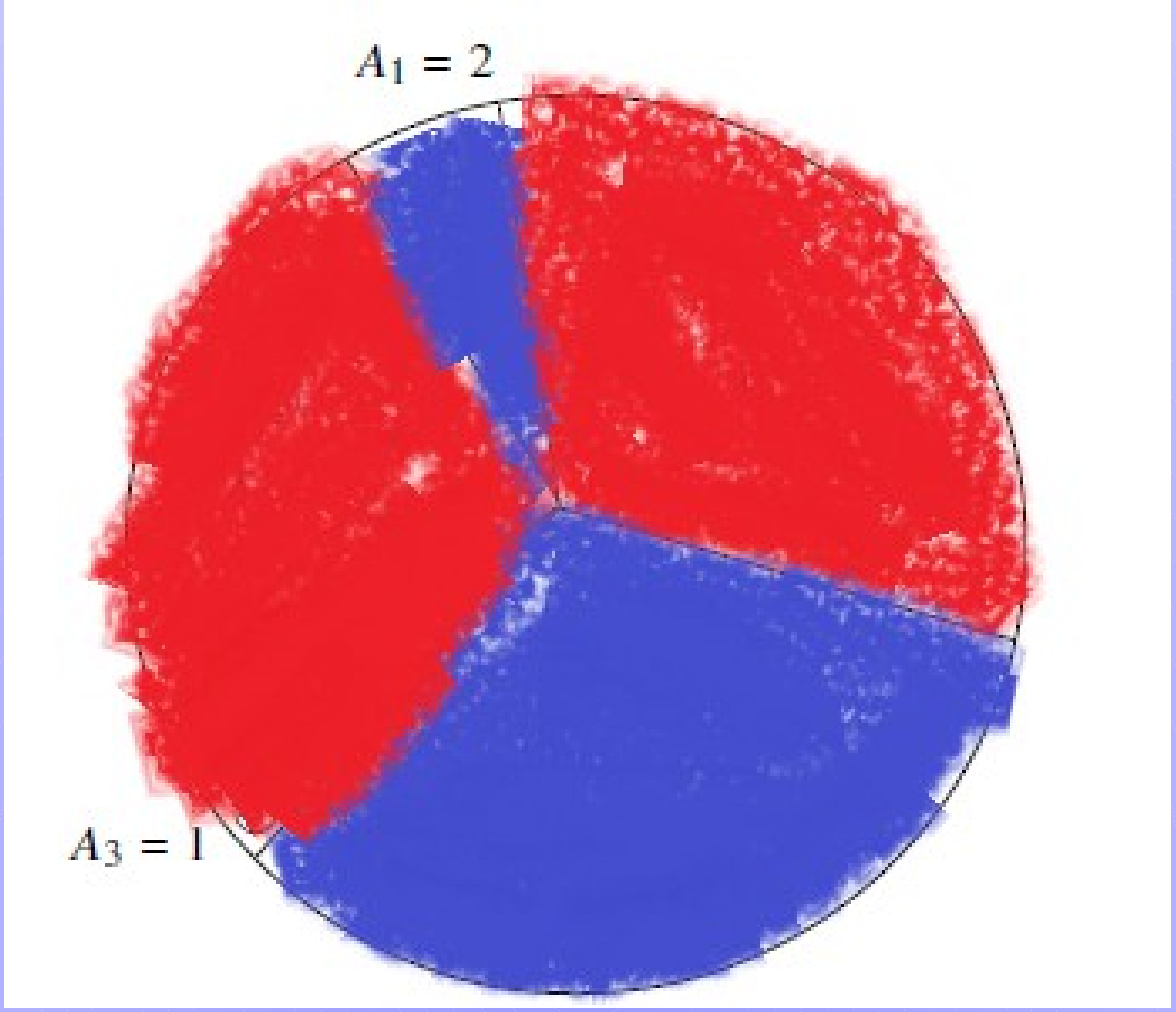
IOI CHAN

12

例

JOI KUN

18

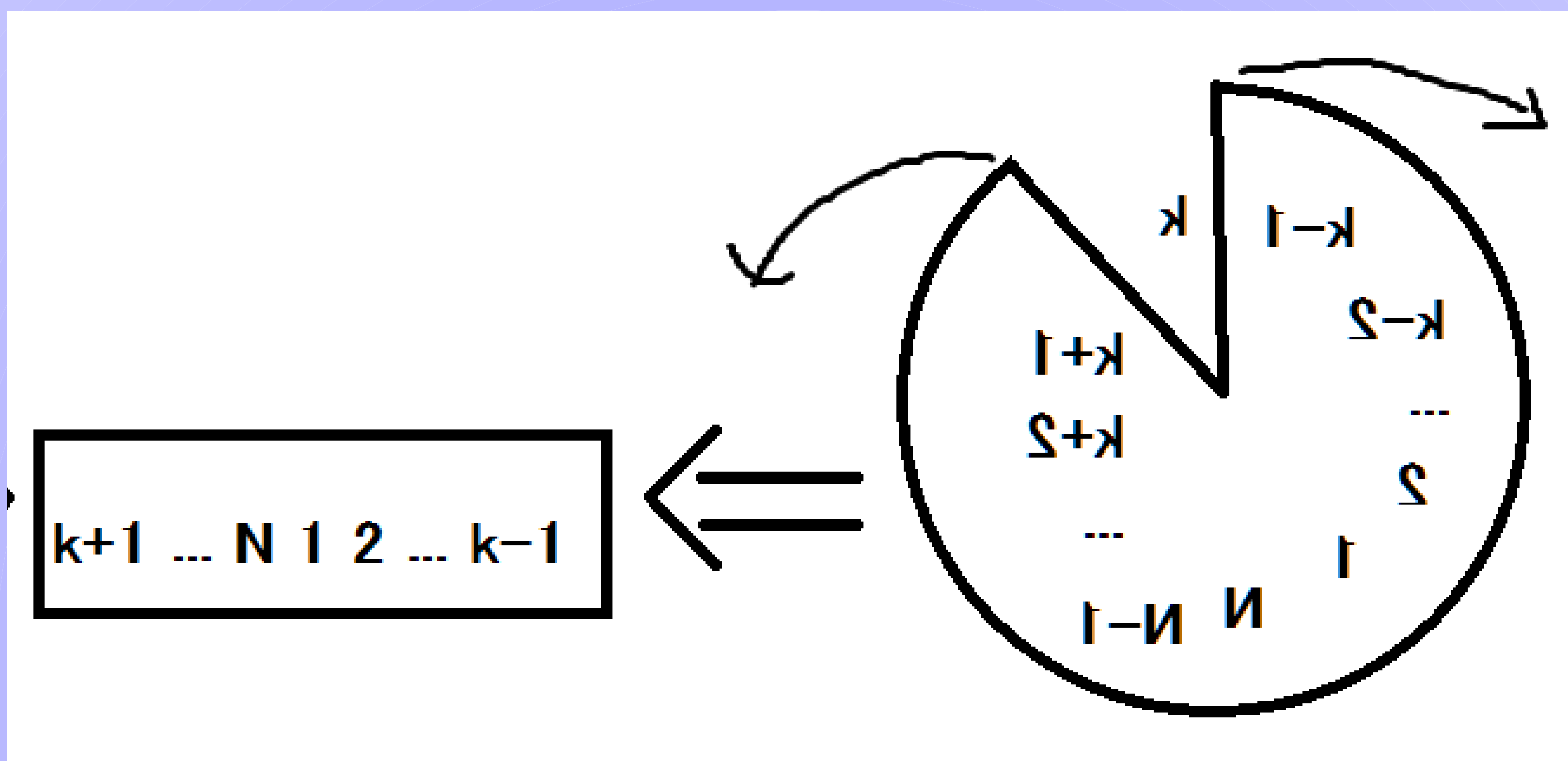


IOI CHAN

12

とりあえず円環は複雑なので、

- JOIくんが最初にとるケーキを全探索
- 残りのケーキは一直線上におけるとしてよい



小課題1

- JOIくんが両方のケーキを取れるときにどっちを取ったかを全探索
- JOIくんがケーキを取るのはN/2回程度
- IOIちゃんの選択は分岐しない(大きいほう)
- 計算量$O(2^{N/2})$くらい　15点

状態をまとめる

- 全探索は無駄が多い
- たとえば、[i,j]の外側でどんな風に取っていっても「残り[i,j]」の状況からの取り方には影響しない

状態をまとめる

- 全探索は無駄が多い
- たとえば、[i,j]の外側でどんな風に取っていっても「残り[i,j]」の状況からの取り方には影響しない
- 動的計画法が使える好条件では？

状態をまとめる

- 全探索は無駄が多い
- たとえば、[i,j]の外側でどんな風に取っていっても「残り[i,j]」の状況からの取り方には影響しない
- 動的計画法が使える好条件では？
- というかあの図を見たら区間DPっぽいよね
- 典型

動的計画法パート

- dp[i][j] :=残り[i,j]の状況からはじめたときのJOIくんの取り分の最大値
- 「残り[i,j]の状況からはじめたとき」がJOIくんとIOIちゃんどっちのターンから始まるかは、区間のながさの偶奇できまる(ので一意)
- なのでdp[i][j][0], dp[i][j][1]に分ける必要はない

動的計画法パート

- 更新はこんな感じの式
- JOIくんのターンから始まる場合

dp[i][j]=

max(dp[i+1][j]+A[i],dp[i][j-1]+A[j]);

- IOIちゃんのターンから始まる場合

dp[i][j]=dp[i+1][j] (A[i]>A[j]のとき)

dp[i][j]=dp[i][j-1] (A[i]<A[j]のとき)

小課題2　まとめ

- JOIくんの最初にとるものを決める(O(N))
- 状態数O(N^2)のDPをする
- (メモ化再帰で書くと楽)

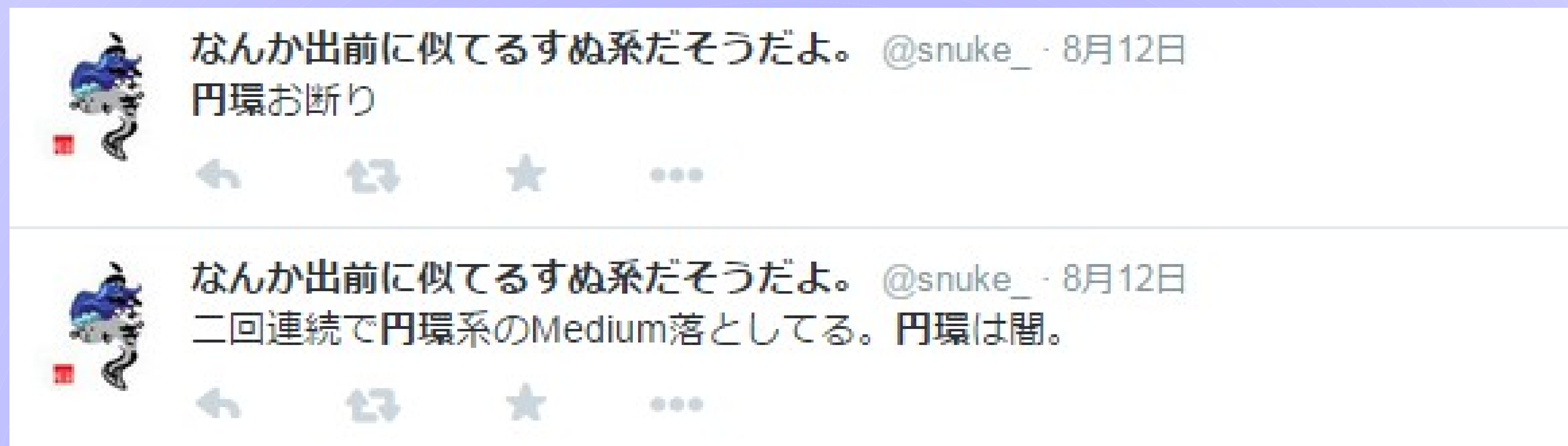
- 全部でO(N^3) 55点

満点解法

- ここまできたらもう余興
- JOIくんの最初の選び方ごとにわざわざいままでのメモしたものを消す必要はない
- ので、状態数は合計でO(N^2)
- 計算量もO(N^2)
- 100点

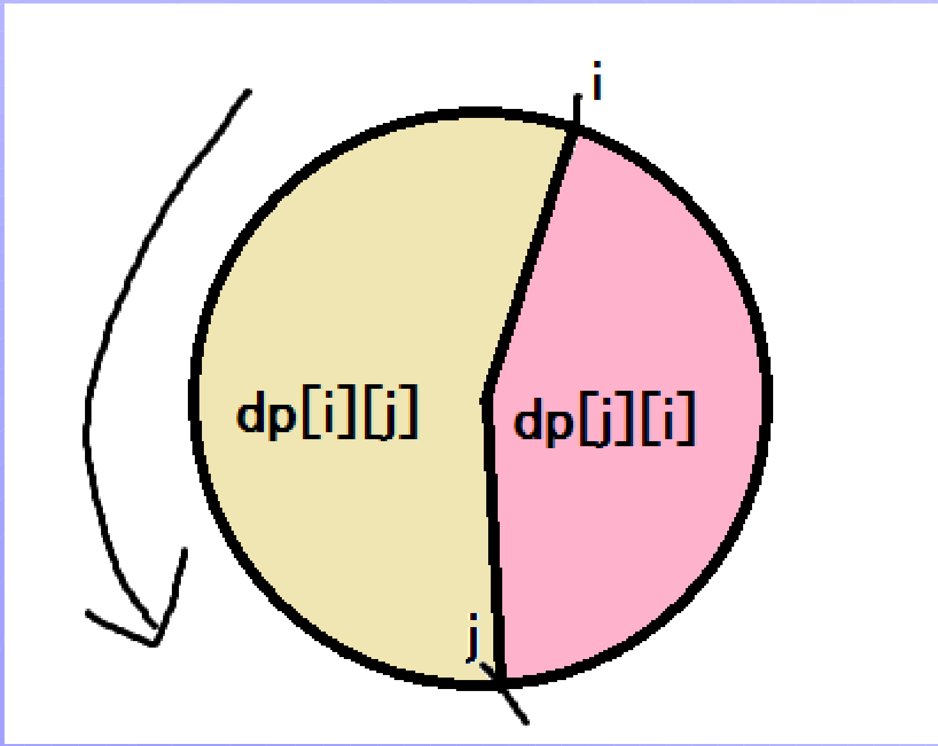
円環

- 円環は実装が結構ややこしいと言われます



実装法(1)

- 本当にdp[i][j]はiから始まってjで終わる区間として、そのまま持つ。
- こんなイメージ



実装法(2)

- 1番目からN番目を一列に順に2回ならべる
- そうすると円環を区間につぶしたまま作業できる
- dp[1][N-1], dp[2][N], dp[3][N+1], …, dp[N][N+N-2]
  を計算していくことになる
- 定数に2や4がかかるけど間に合う

得点分布