警備員（Security Guard）

解説　nxteru（戸高空）

はじめに

この解説の多くの部分で証明を飛ばしています

[チューターのnoshi91さんによるありがたい証明]

問題概要

- N頂点のグラフ
- ある2頂点に停泊できそれらを行き来するM隻の船

セキュリティ条件

頂点vに停泊する船には常にS[v]人以上の警備員がいる

問題概要

小問題

- 最初、いくつかの船を廃止、残っている船をどちらかに停泊させ好きな人数の警備員を乗せる
- 次の条件を満たすために必要な警備員の最小人数を求めよ

問題概要

条件

どの頂点u,vについてもセキュリティ条件を満たしながら次の行動を繰り返して客をuからvまで輸送できる

- 停泊する船と停泊する頂点の間で警備員または客が乗り降りする
- ある頂点に停泊する船がもう一方に移動し停泊

問題概要

小問題を次のk=0,1,2,⋯,Qの場合について解け

- 初めに船を新たにk隻増やすことができる
- ただしそれらの船が行き来する2つの頂点は自由に選ぶことができる

制約

- N<=200000
- M<=400000
- Q<=200000
- 1<=S[v]<=10^9（1<=v<=N）

小課題

1. （12点） Q=0,パス,S[v]<=2
2. （13点） Q=0,パス
3. （12点） Q=0,木
4. （13点） Q=0
5. （8点） N<=16
6. （18点） N<=3000
7. （24点） 追加の制約はない

まずはQ=0(小問題)までを解く

1. （12点） Q=0,パス,S[v]<=2
2. （13点） Q=0,パス
3. （12点） Q=0,木
4. （13点） Q=0

まずはQ=0(小問題)までを解く

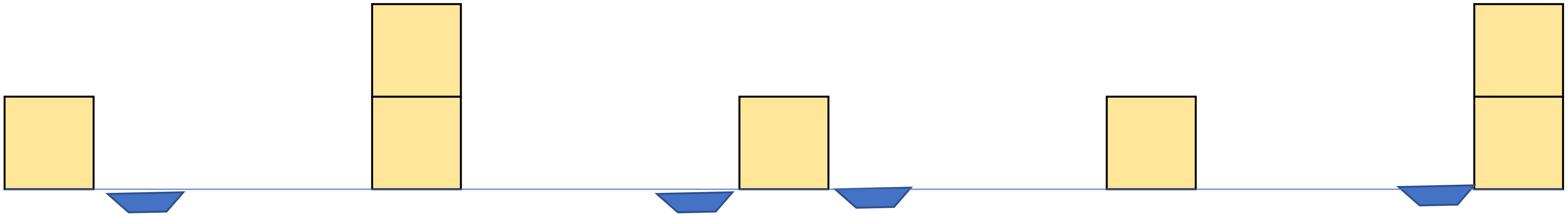
小課題3の木を解くまでの方針を1つ紹介

s[v]の最小値で全部埋めて消すを繰り返す

s[v]の最小値で全部埋めて消すを繰り返す

小課題1. Q=0,パス,S[v]<=2

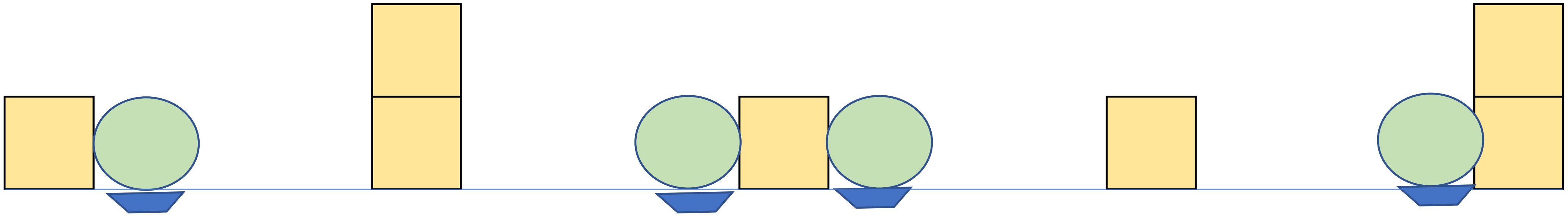
- S[v]は1か2



s[v]の最小値で全部埋めて消すを繰り返す

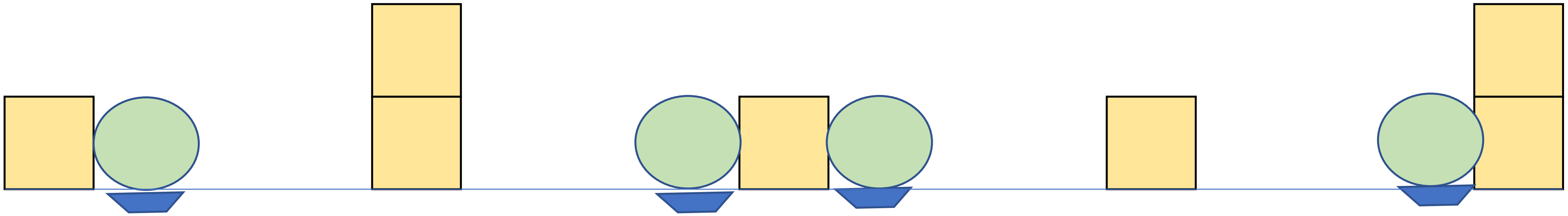
小課題1. Q=0,パス,S[v]<=2

- S[v]は1か2
- とりあえずすべての船に1人は必ず必要



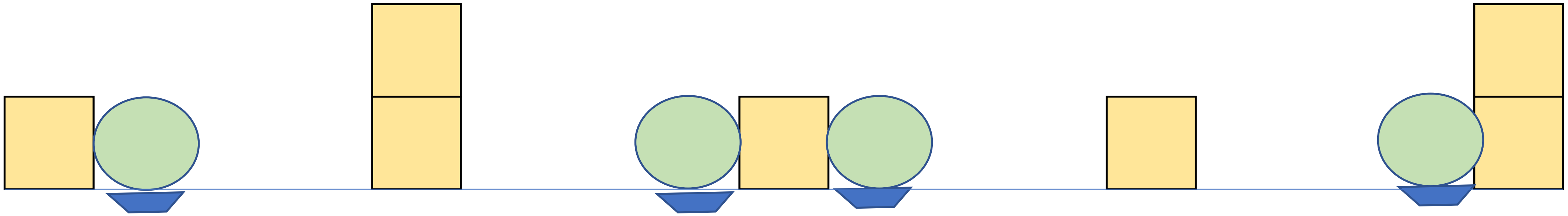
s[v]の最小値で全部埋めて消すを繰り返す

- このときその1人はずっとその船に常駐しているとしてよい



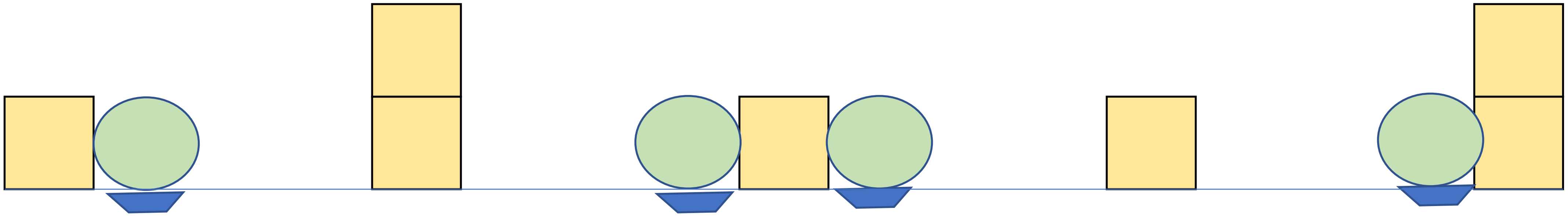
s[v]の最小値で全部埋めて消すを繰り返す

- このときその1人はずっとその船に常駐しているとしてよい
- すべての船について1人は必ずいる状態



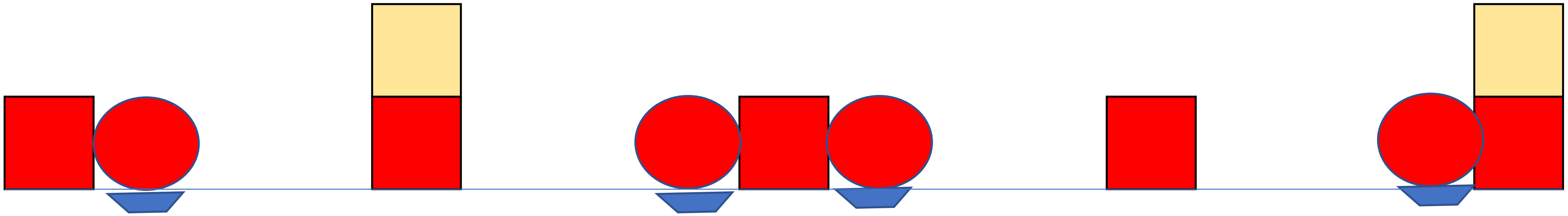
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のS[v]が1下がったと考えることができる



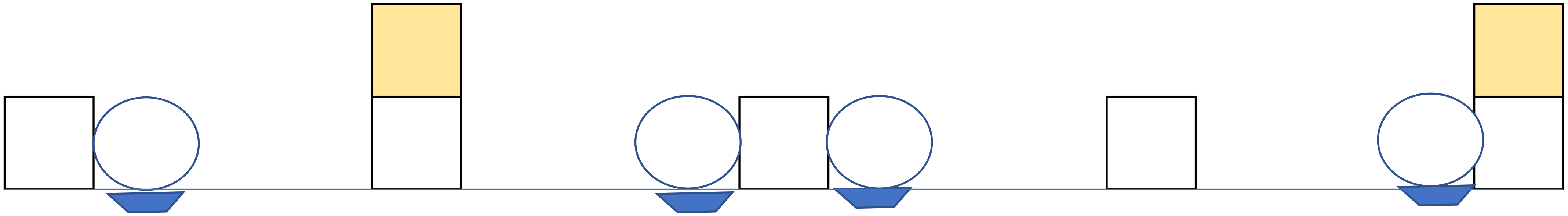
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のS[v]が1下がったと考えることができる



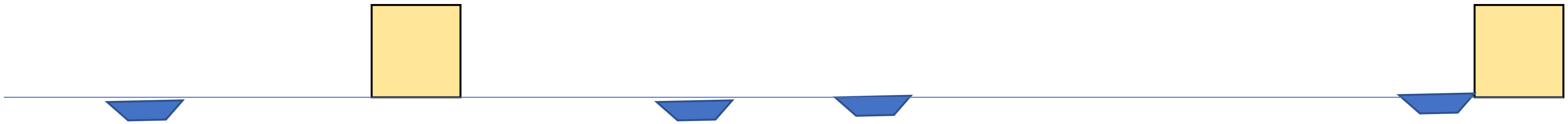
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のS[v]が1下がったと考えることができる



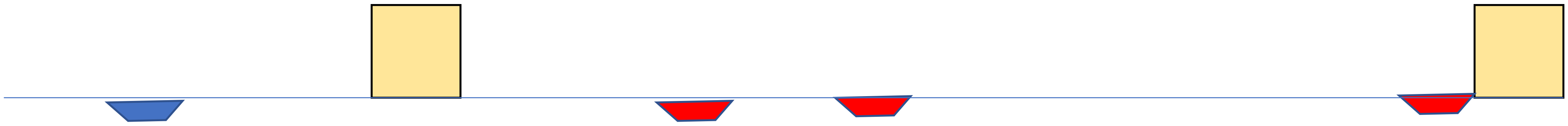
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のS[v]が1下がったと考えることができる



s[v]の最小値で全部埋めて消すを繰り返す

- S[v]が0になったところはその上を警備員が自由に移動できるので、その間の船は合わせて1つとみなす



s[v]の最小値で全部埋めて消すを繰り返す

- S[v]が0になったところはその上を警備員が自由に移動できるので、その間の船は合わせて1つとみなす

s[v]の最小値で全部埋めて消すを繰り返す

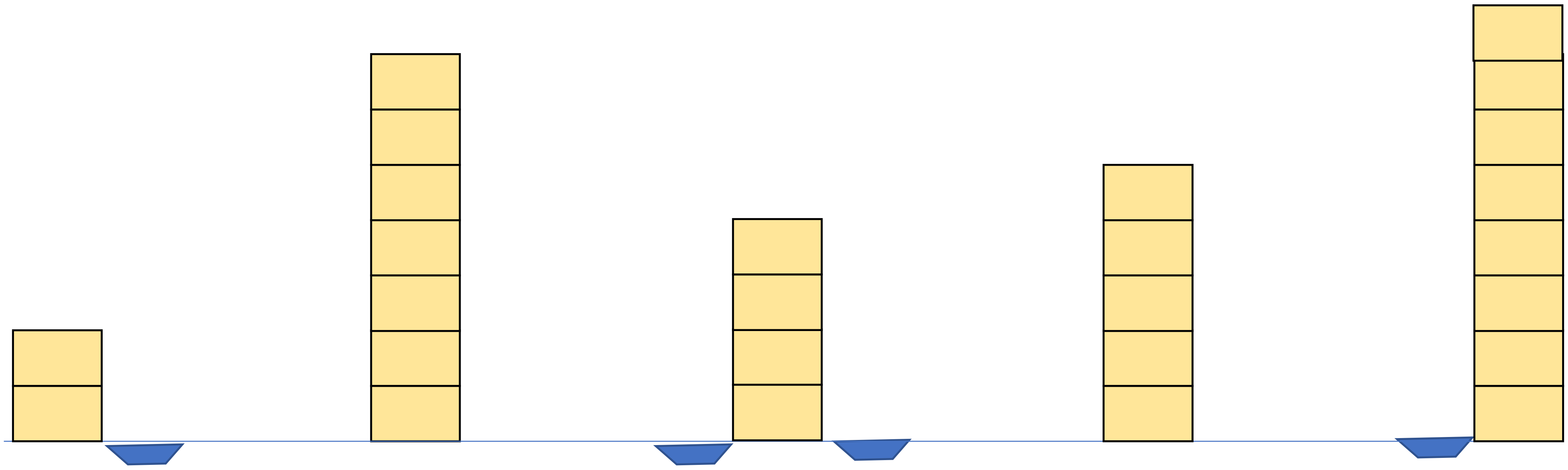
- これをあと1回すればよい

s[v]の最小値で全部埋めて消すを繰り返す

- これで求められる人数以上は必ず必要
- またこれで求められた人数がいれば自由に輸送できる

s[v]の最小値で全部埋めて消すを繰り返す

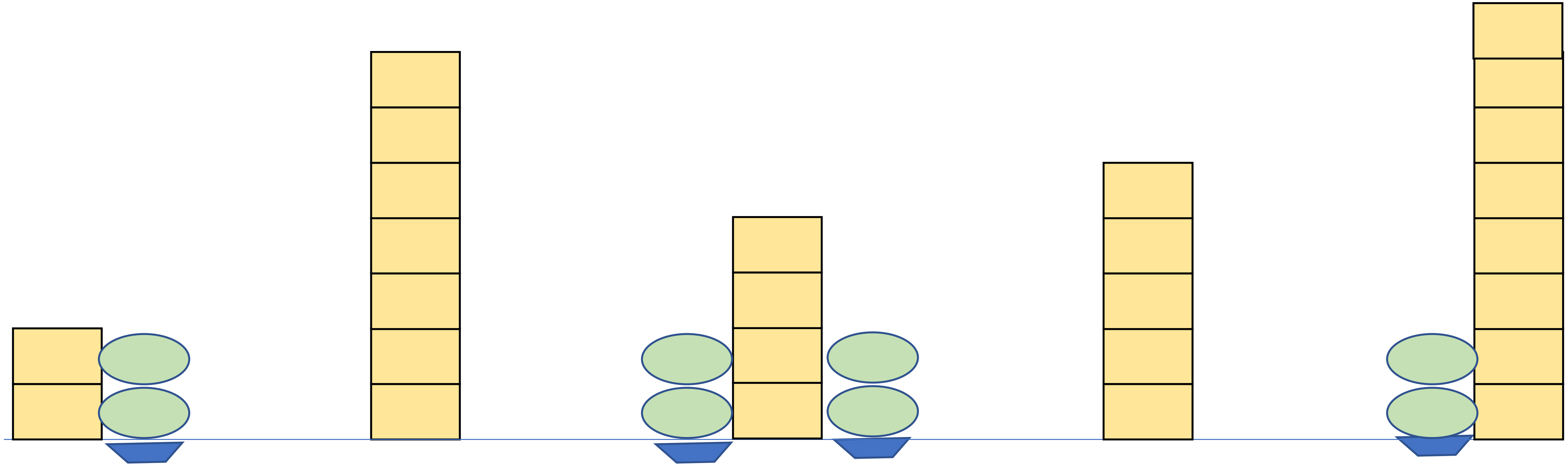
小課題2. Q=0,パス



s[v]の最小値で全部埋めて消すを繰り返す

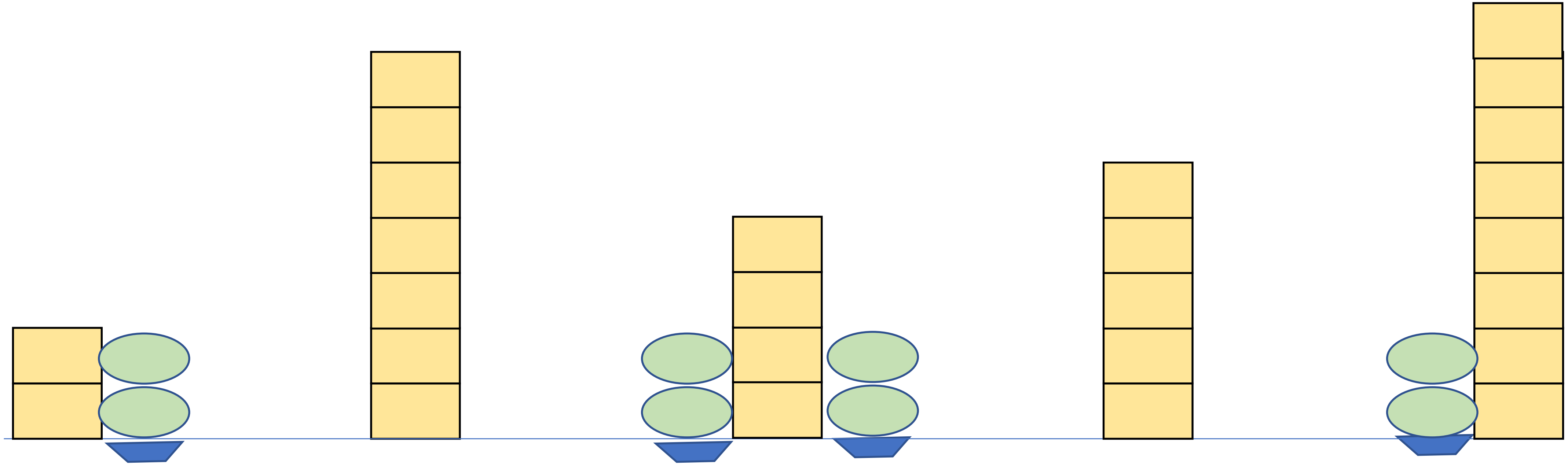
小課題2. Q=0,パス

- とりあえずすべての船にSの最小値の人数は必要



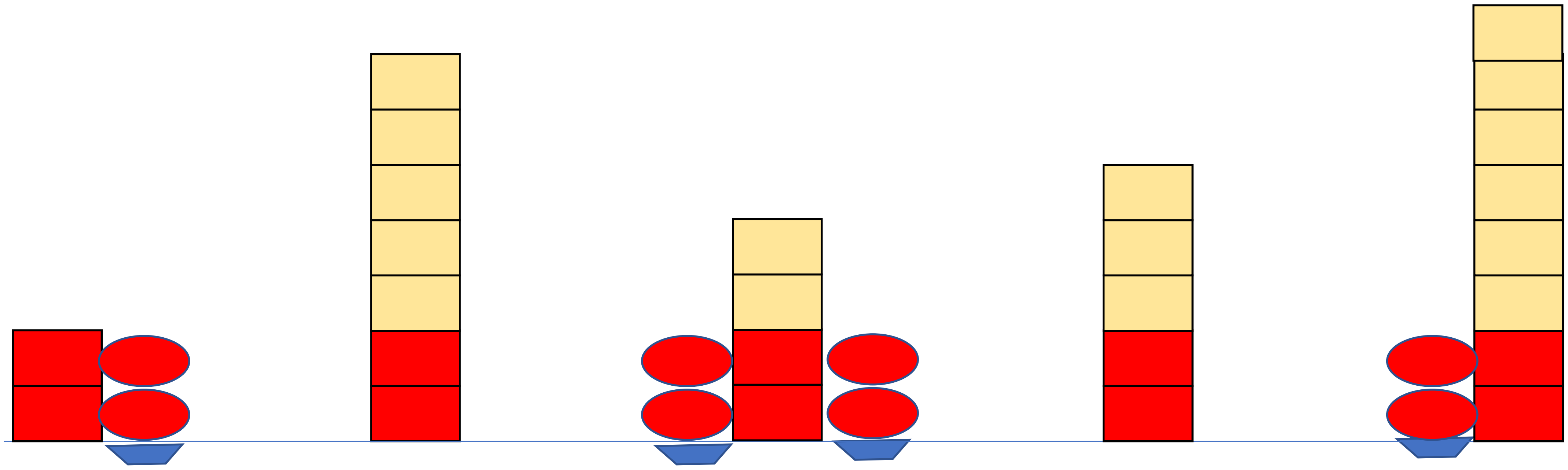
s[v]の最小値で全部埋めて消すを繰り返す

- その人はずっとその船に常駐しているとしてよい
- すべての船について最小値人は必ずいる状態



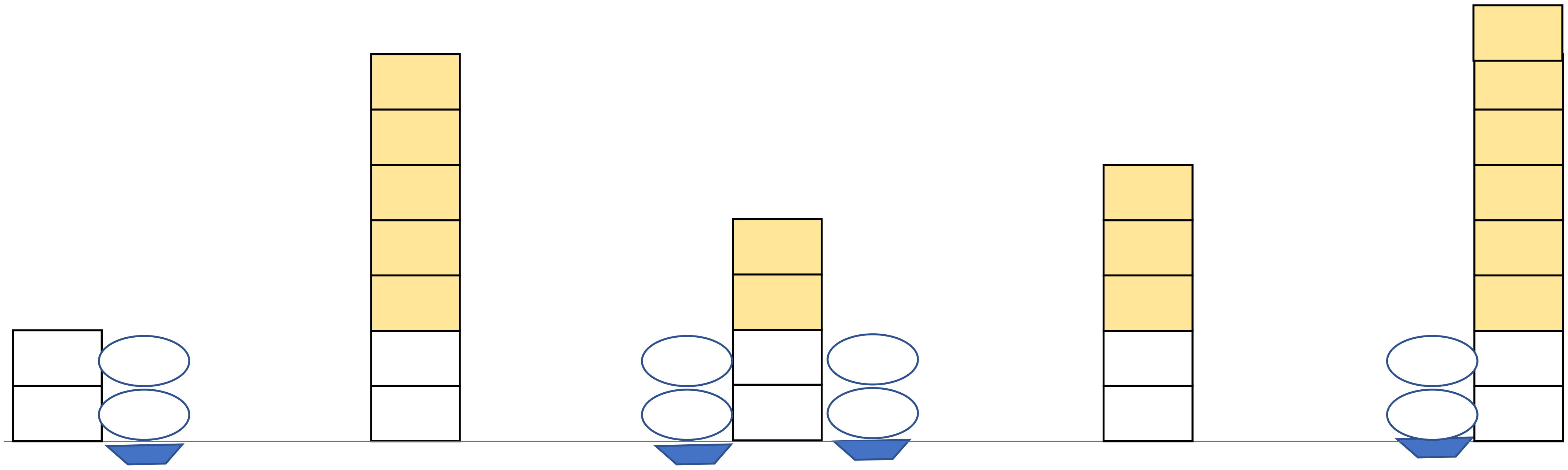
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のSが下がったと考える



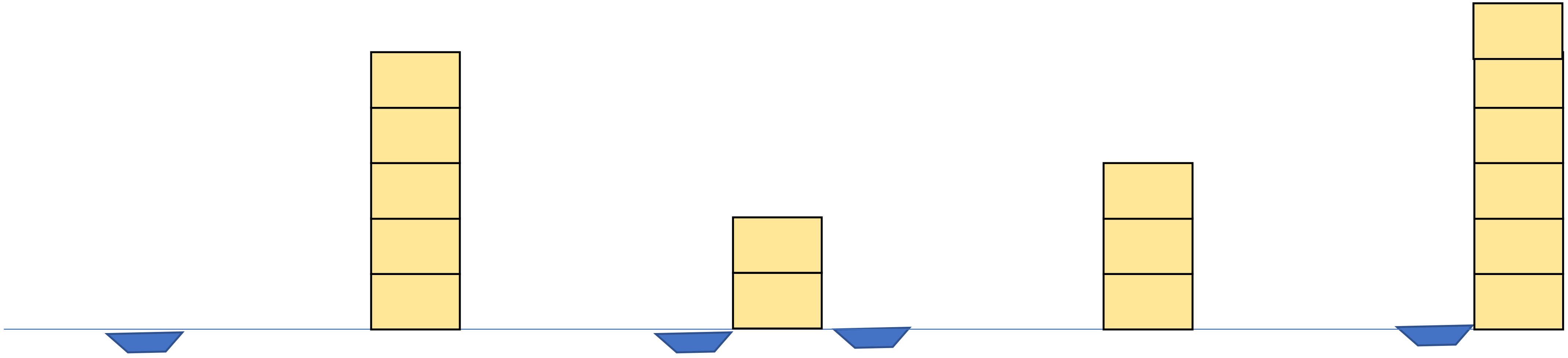
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のSが下がったと考える



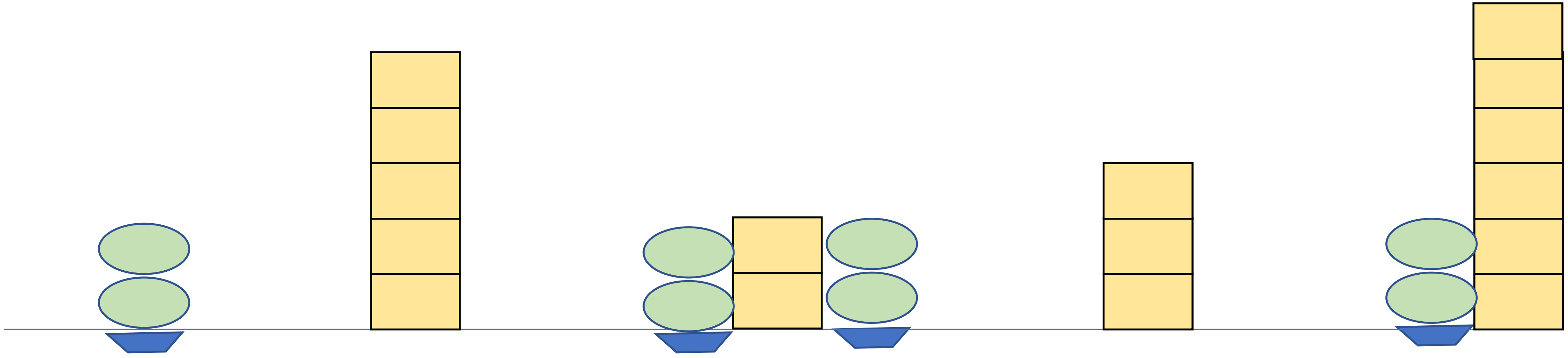
s[v]の最小値で全部埋めて消すを繰り返す

- 全体のSが下がったと考える



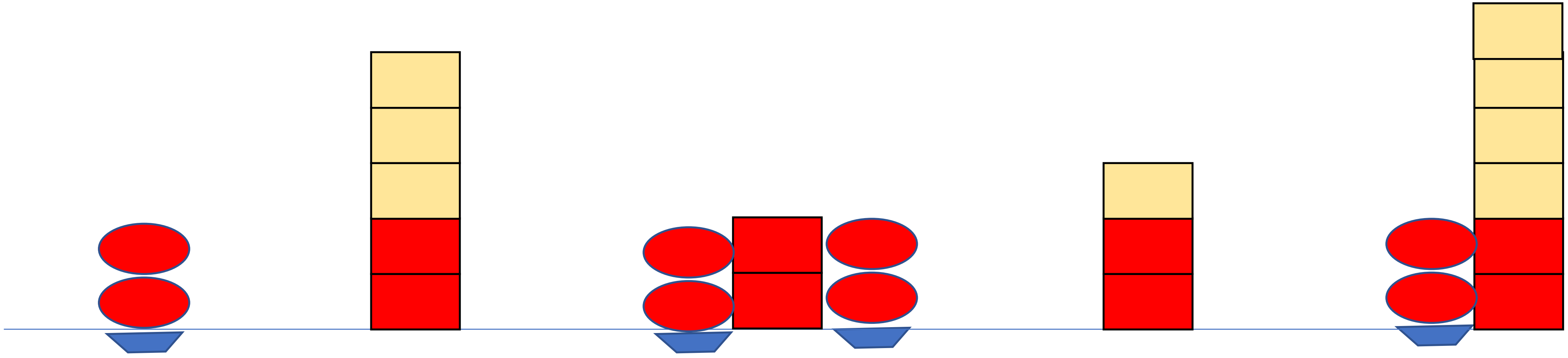
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす



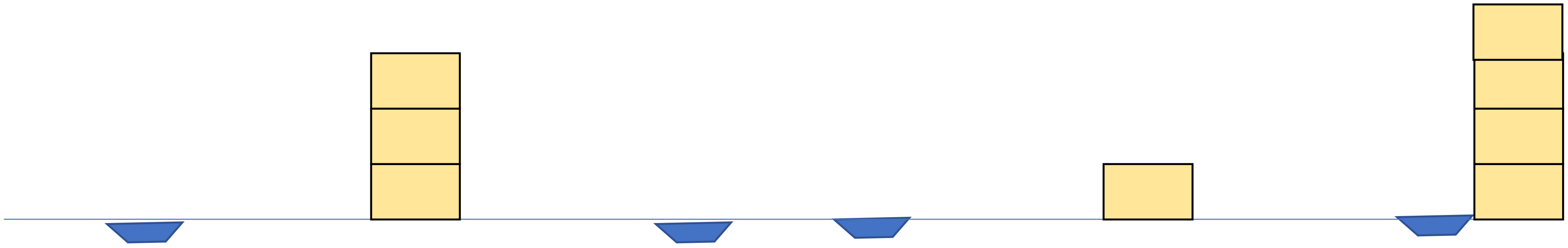
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす



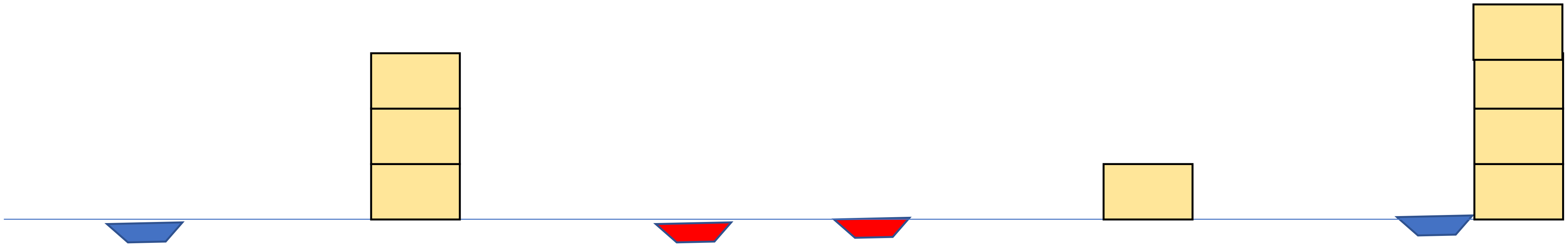
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす



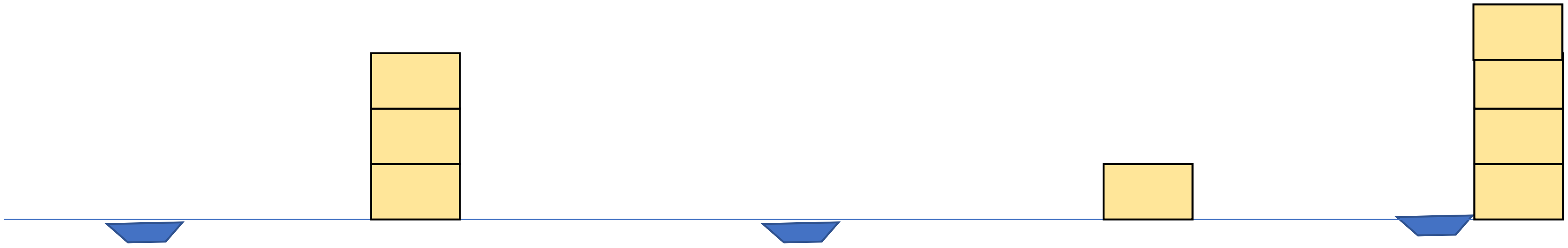
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる



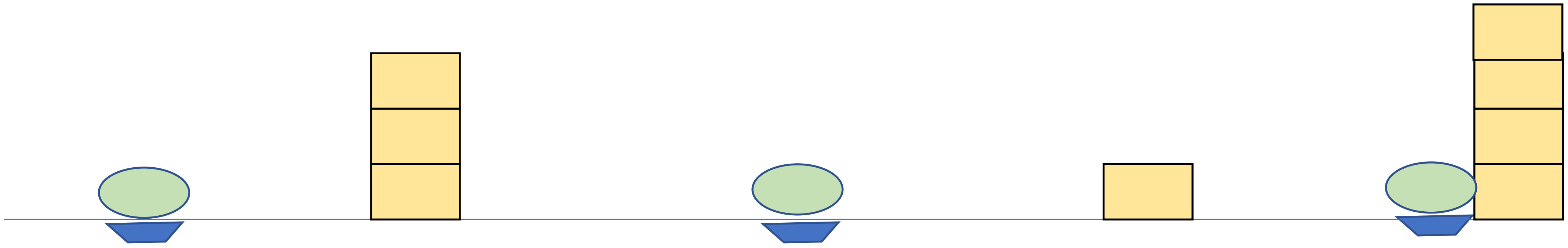
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる



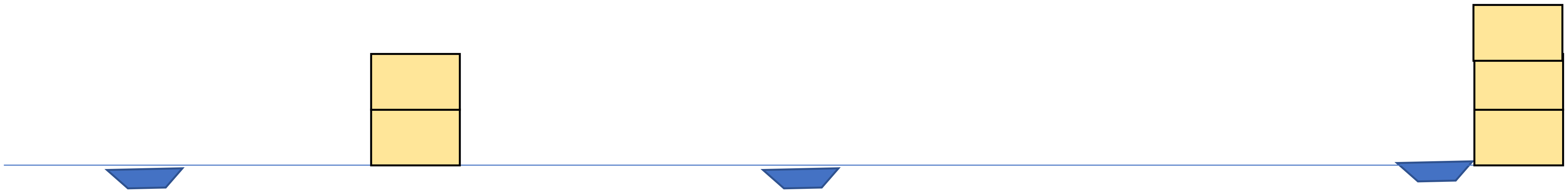
s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる



s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる



s[v]の最小値で全部埋めて消すを繰り返す

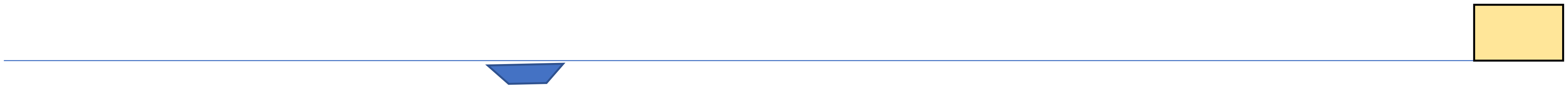
- これをくりかえす
- 頂点がなくなったら船を合わせる

s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる

s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる



s[v]の最小値で全部埋めて消すを繰り返す

- これをくりかえす
- 頂点がなくなったら船を合わせる

s[v]の最小値で全部埋めて消すを繰り返す

- ここまでに足した警備員の数の合計を答えればよい

s[v]の最小値で全部埋めて消すを繰り返す

- シミュレーションしてもよいが整理するとこの人数は次のように定式化できる

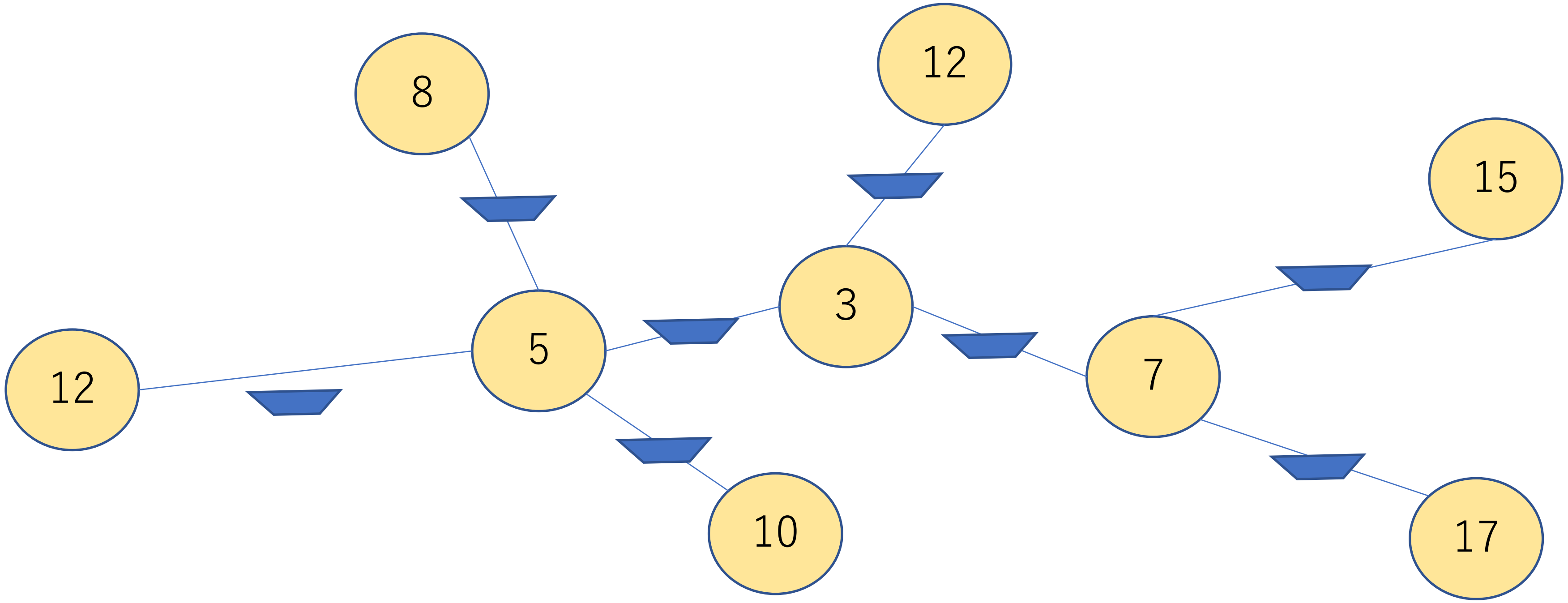
S[2]+S[3]+⋯+S[N-1]+max(S[1],S[2]⋯,S[N])

s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：8
合計：0

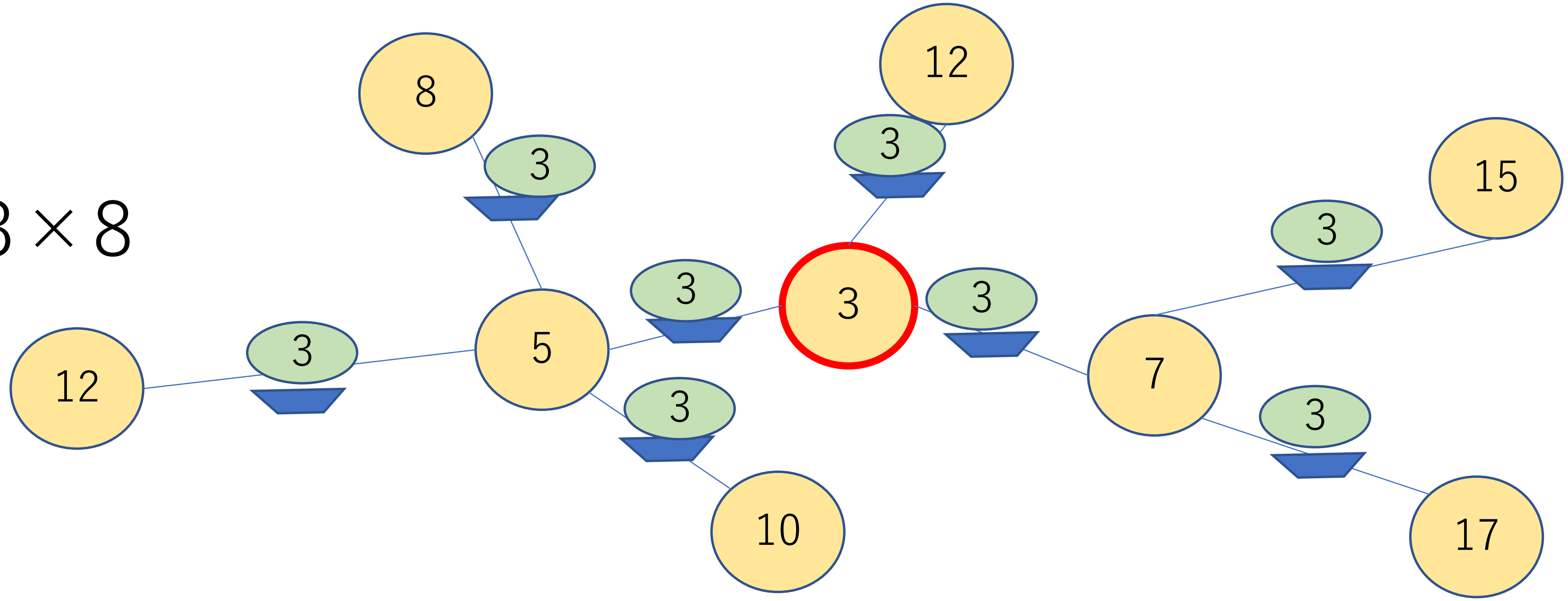


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：8
合計：0+3×8

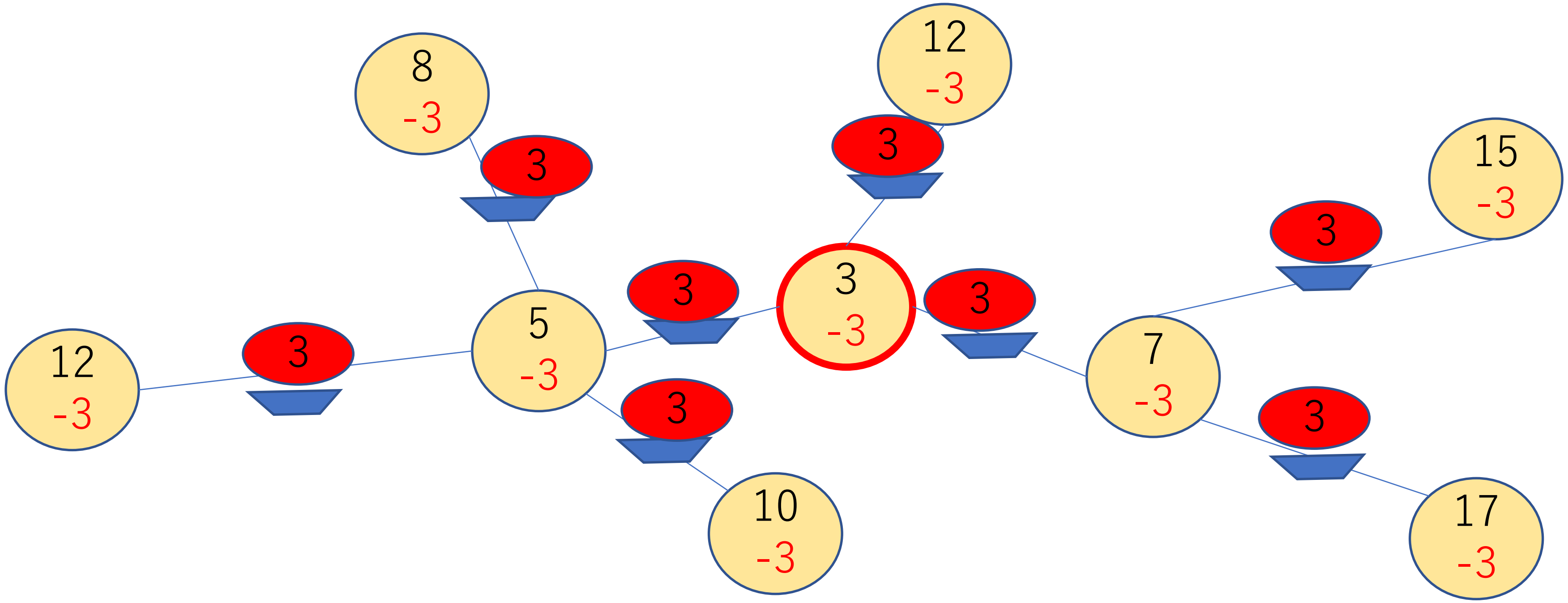


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

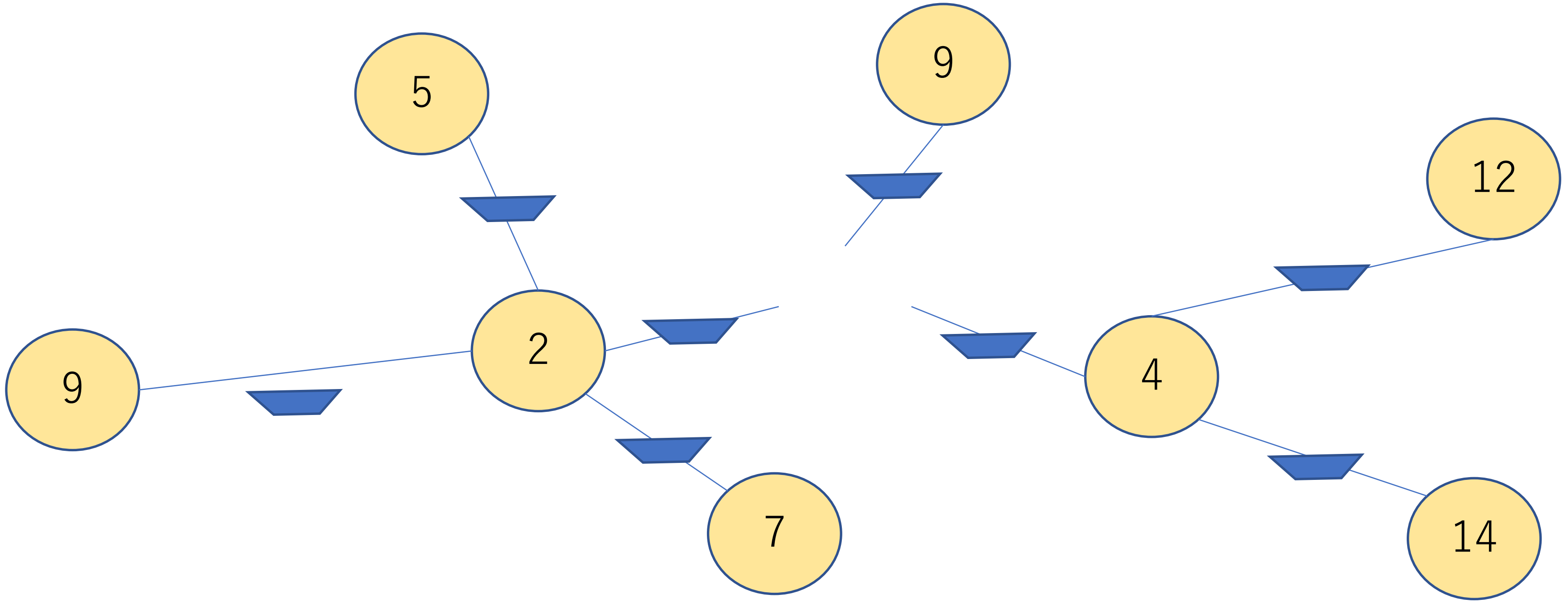
船の数：8
合計：24



s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木
- この考え方を木に拡張する

船の数：8
合計：24

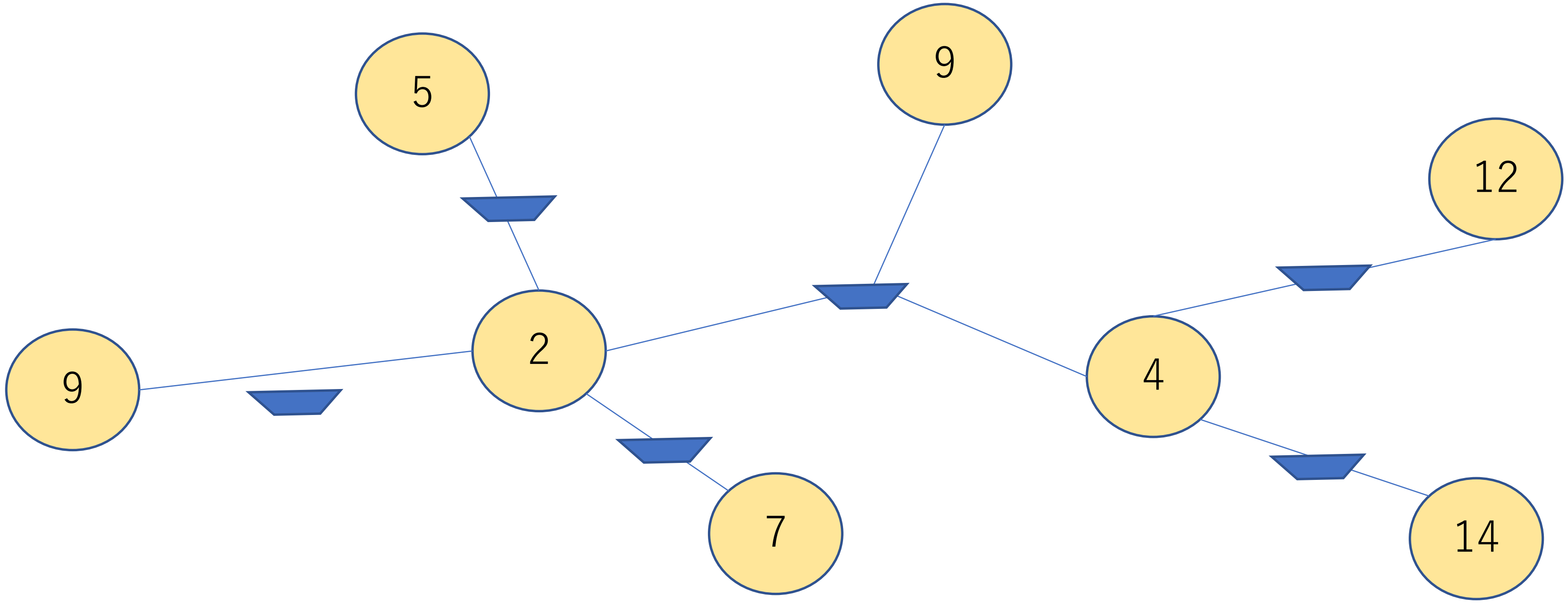


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：6
合計：24

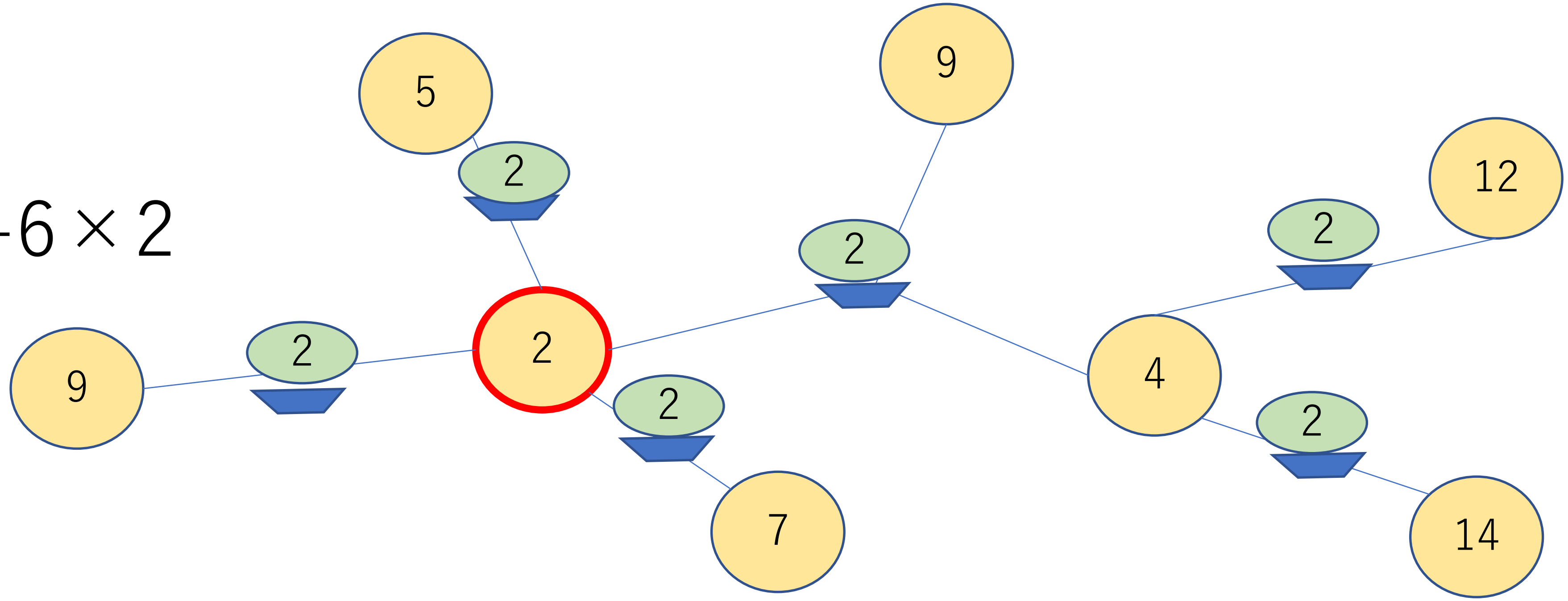


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：6
合計：24+6×2

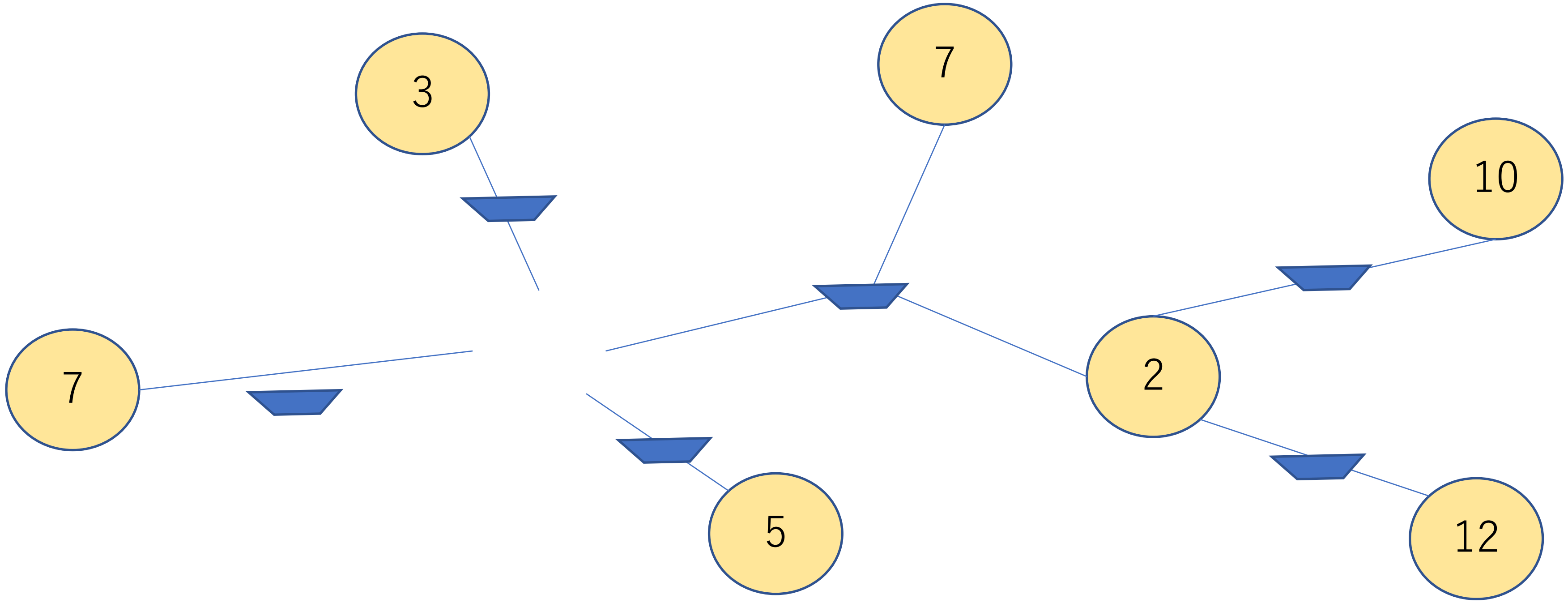


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：6
合計：36

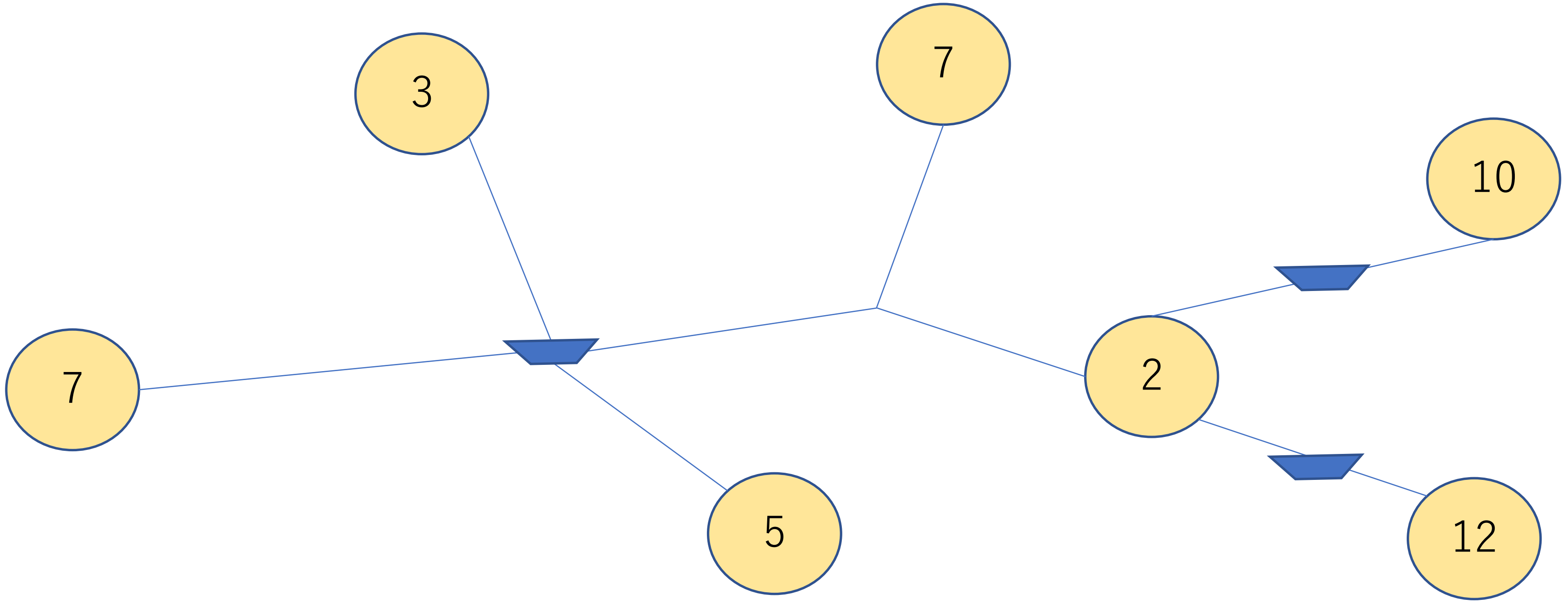


s[v]の最小値で全部埋めて消すを繰り返す

小課題3. Q=0,木

- この考え方を木に拡張する

船の数：3
合計：36



s[v]の最小値で全部埋めて消すを繰り返す

- 頂点をSでソート、小さい順に消す
- i番目に小さい頂点を消すとき
  – 警備員の合計に+(S[i]-S[i-1])×(船の数)
  – 船の数をdeg[i]だけ減らす（deg[i]は頂点iの次数）

- 木の形は関係なくdeg[i]とS[i]だけで計算できる

s[v]の最小値で全部埋めて消すを繰り返す

- さきほどの計算を整理すると次のように定式化できる

$\Sigma S[i] \times (deg[i]-1)+max(S)$

小課題4 Q=0（一般のグラフ）

- 木の制約がなくなって一般のグラフになった
- **使う辺が木であるような最小値を達成するやり方が存在する**

- 証明は省略

小課題4 Q=0（一般のグラフ）

- 木を定めたときのコスト$\Sigma S[i] \times (deg[i]-1)+\max(S)$
- 上の値が最小になるような木を選ぶ問題
- 上の式$=\Sigma_{(x, y) \in E} S[x]+S[y] - \Sigma S[i] +\max(S)$

(各辺についてコストを両端のSの和としたとき、辺のコストの合計-Sの合計+max(S))

となるから辺(x,y)のコストをS[x]+S[y]とした最小全域木を求めればよい

小課題4 (Q=0)までまとめ

- 各辺(x,y)についてコストがS[x]+S[y]の最小全域木を求める

小課題5 以降

- 辺(x,y)のコストがS[x]+S[y]のときの最小全域木を次のk=0,1,2,⋯,Qの場合について求める

- あらたに好きな2頂点を選んで辺を追加することをk回できる

小課題5 以降

5. (8点）　N<=16
6. (18点）N<=3000
7. (24点）追加の制約はない

小課題5 以降

- もともとあった辺で使うのはk=0の時求めた最小全域木の辺のみ
- 新たに追加した辺は必ず使う

- **もとの最小全域木からk本消して、新たにk本つなぐとしてよい**
- もとの最小全域木の辺を**元辺**、新たに追加する辺を**新辺**と呼ぶ

小課題5 以降

- **新たに追加する辺はすべてS最小の頂点を端点とする**

- 消すk本の辺を決め打ったとき，新たにk本追加してすべて連結にするコスト最小化になる
- 各連結成分ごとでS最小の頂点のみ考える
- S最小の頂点と各連結成分をつなぐのが最適

小課題5 (8点）　N<=16

- **新たに追加する辺はすべてS最小の頂点を端点とする**

- 新たに追加する辺の候補がN-1本に絞られたので追加の仕方2^(N-1)通り全探索

小課題6 (18点） N<=3000

- **k=x+1のときの最適な木について、k=xで辺の差が2本（つまり元辺1本、新辺1本の差）であるような最適な木が存在する**

- 証明は略

小課題6 (18点） N<=3000

- k=0,1,2,⋯,Qについて順番に元辺を1本消して，新辺を1本追加する操作を繰り返していく
- このときそれぞれのタイミングでコストが最小になるような操作を貪欲に選んでいけばよい

小課題6 (18点） N<=3000

- k=xが求まっていたときk=x+1をO(N)で求める
- S最小の頂点を根とした根付き木にする
- 消す元辺を全探索するO(N)
- 消す元辺を決め打ったとき，追加する新辺は部分木でS最小の頂点と根を結ぶ辺
- 部分木のS最小値を前計算で求めておくとO(1)

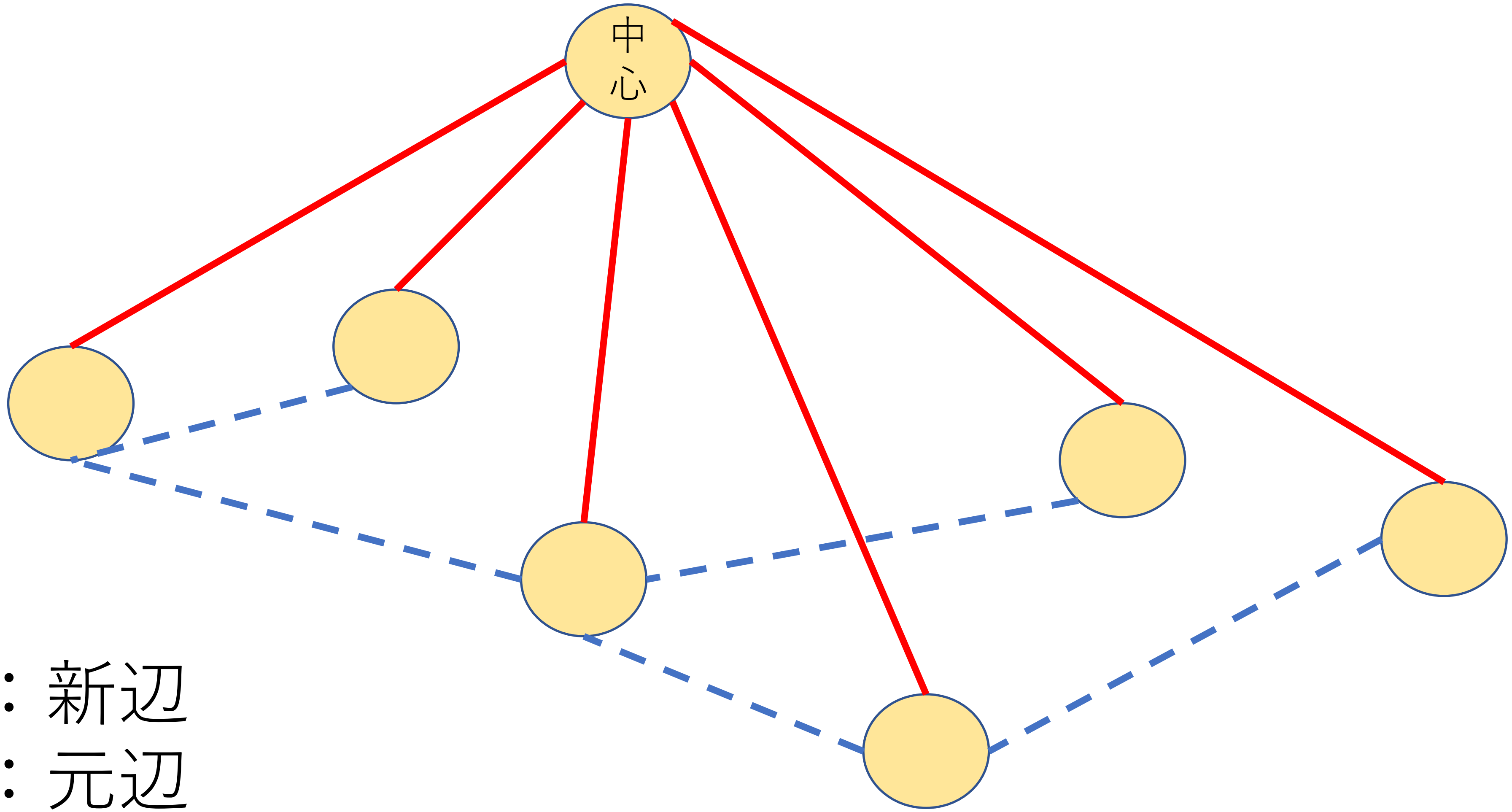
小課題7 (24点)追加の制約なし(N<=200000)

- 辺は消すより追加する方が簡単
- 操作を逆から考える
- S最小の頂点を中心とするスターグラフからスタート
- 最初はすべて中心と各頂点を結ぶ新辺
- 連結性が保たれるように新辺を1本削除して元辺を1本追加するという操作を繰り返す

小課題7 (24点)追加の制約なし(N<=200000)

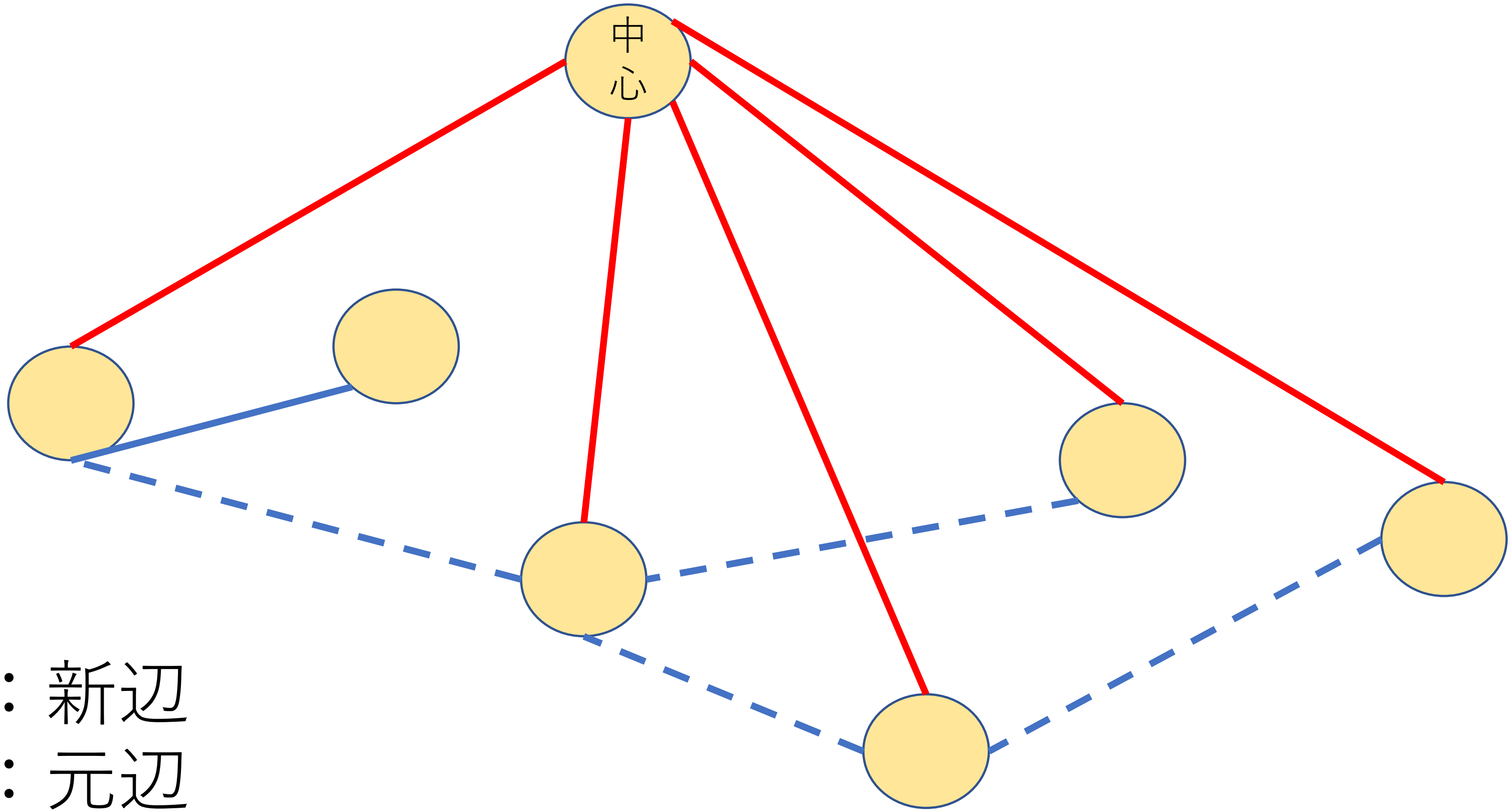
- この場合でも各操作でコストが最小になるものを貪欲に選んでいけばよい

小課題7 (24点)追加の制約なし(N<=200000)



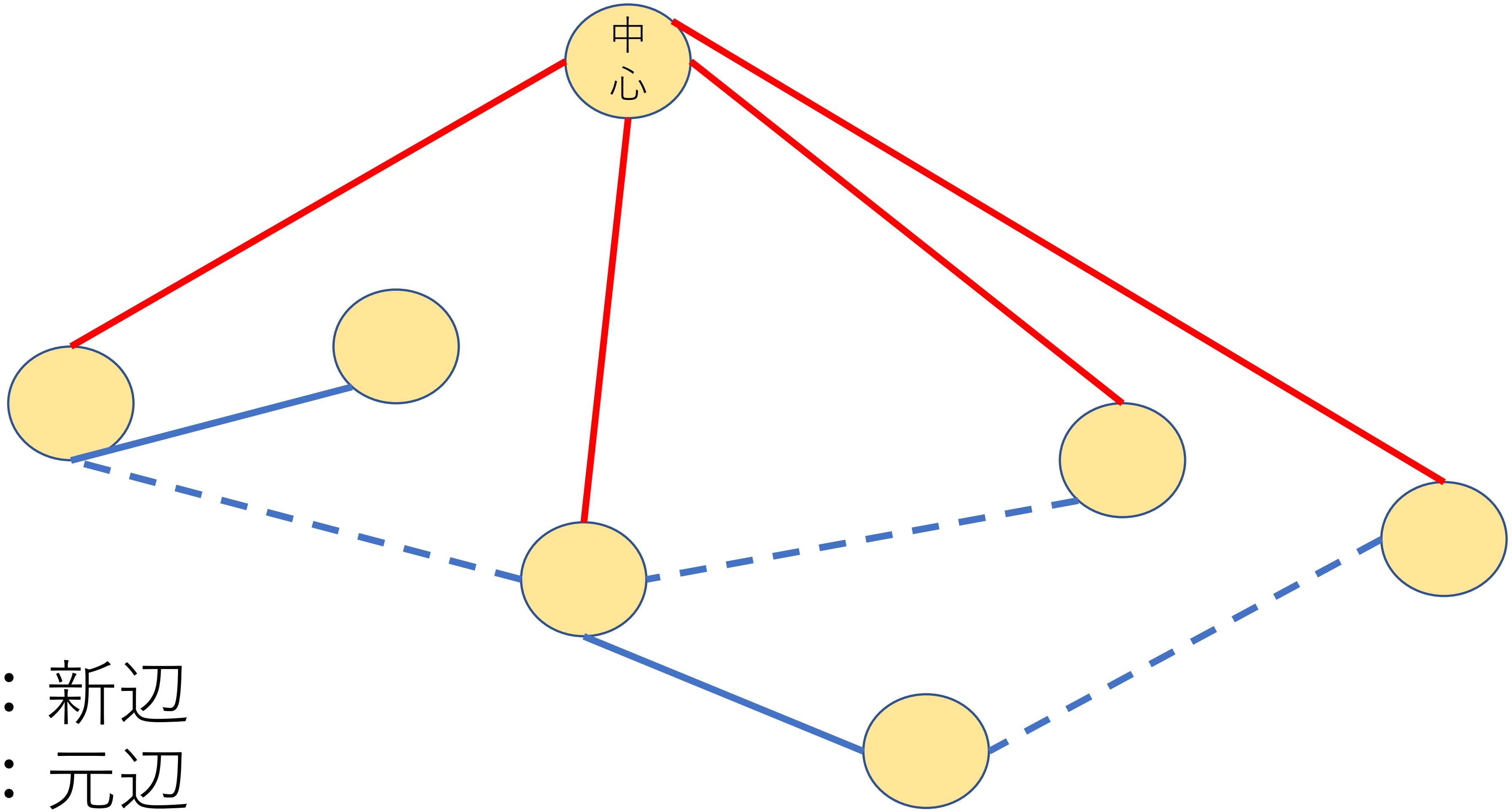
赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)



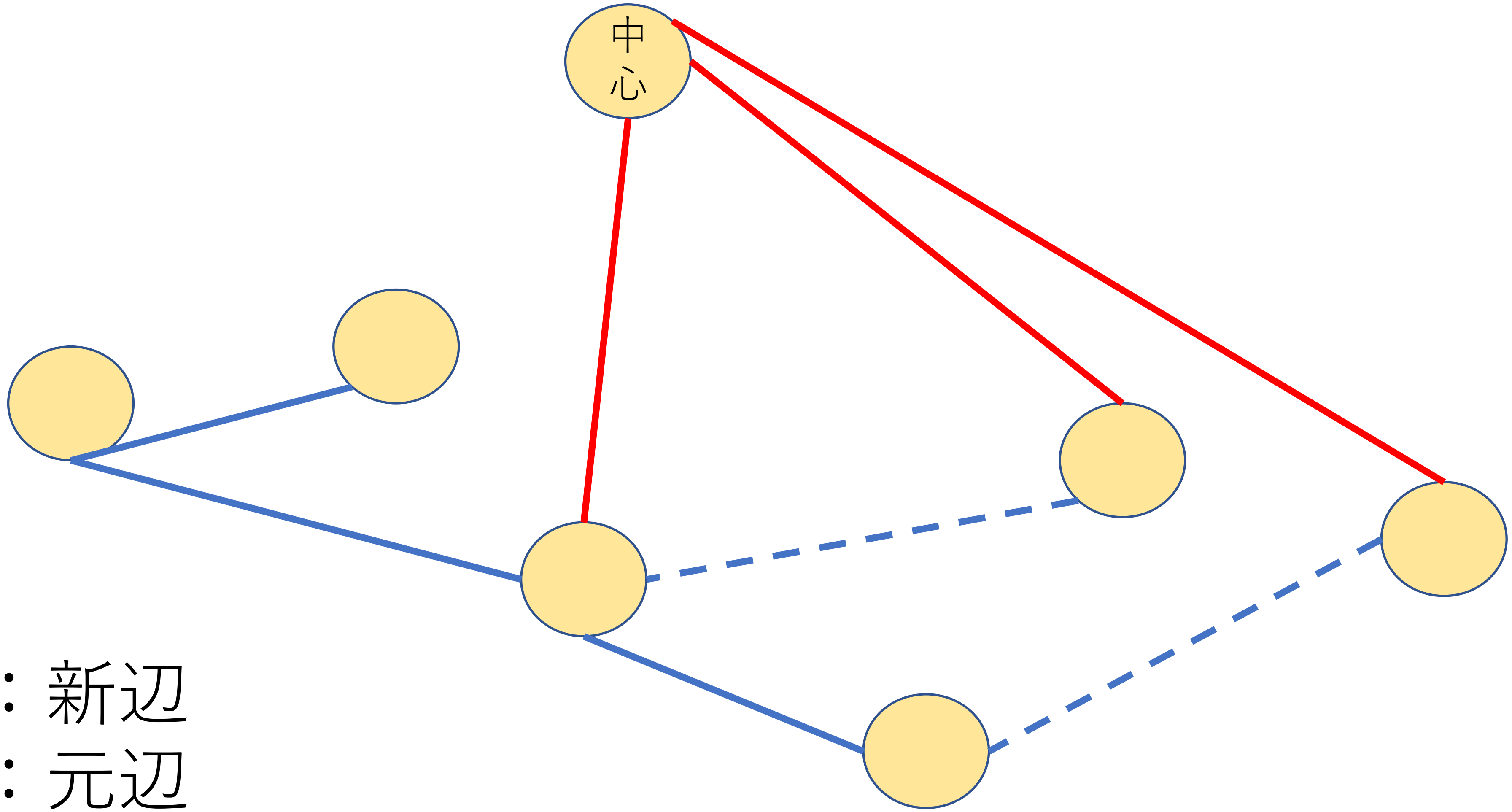
赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)



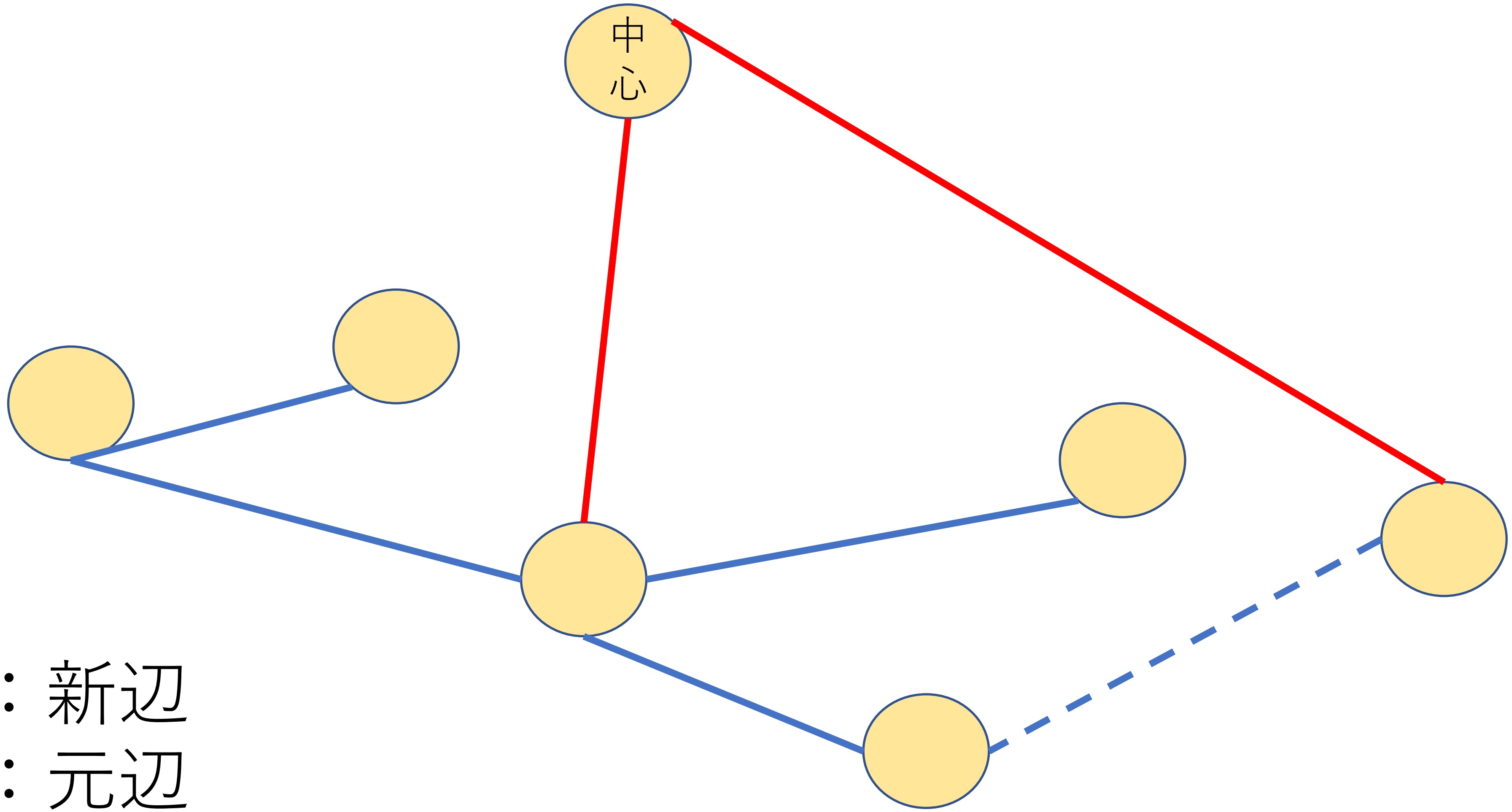
赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)



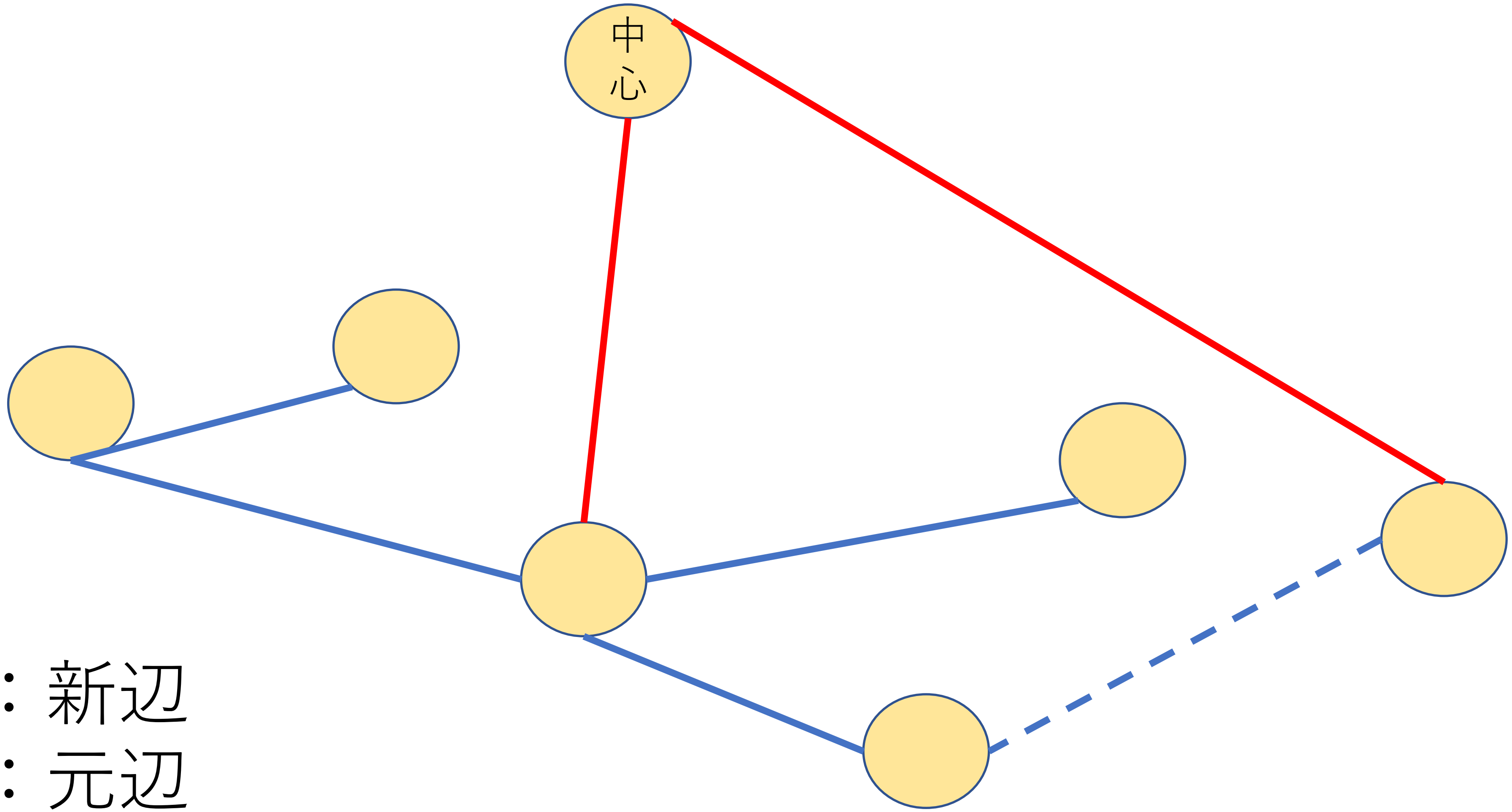
赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)



赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)



赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)

- 各新辺について、それを消すときに追加する元辺は，その新辺がつなぐ連結成分からでている元辺でコスト最小のものとしてよい

- 各新辺を消すとしたときのコストを管理する

小課題7 (24点)追加の制約なし(N<=200000)

- 新辺を消して元辺を追加すると、2つの連結成分がマージされる
- このとき連結成分のもつ元辺がマージされ，その連結成分について新辺を消すときのコストが更新される（その連結成分のもつコスト最小の元辺が変わるから）
- 逆にこれに関わらない新辺はそれを消すときのコストは変化しない

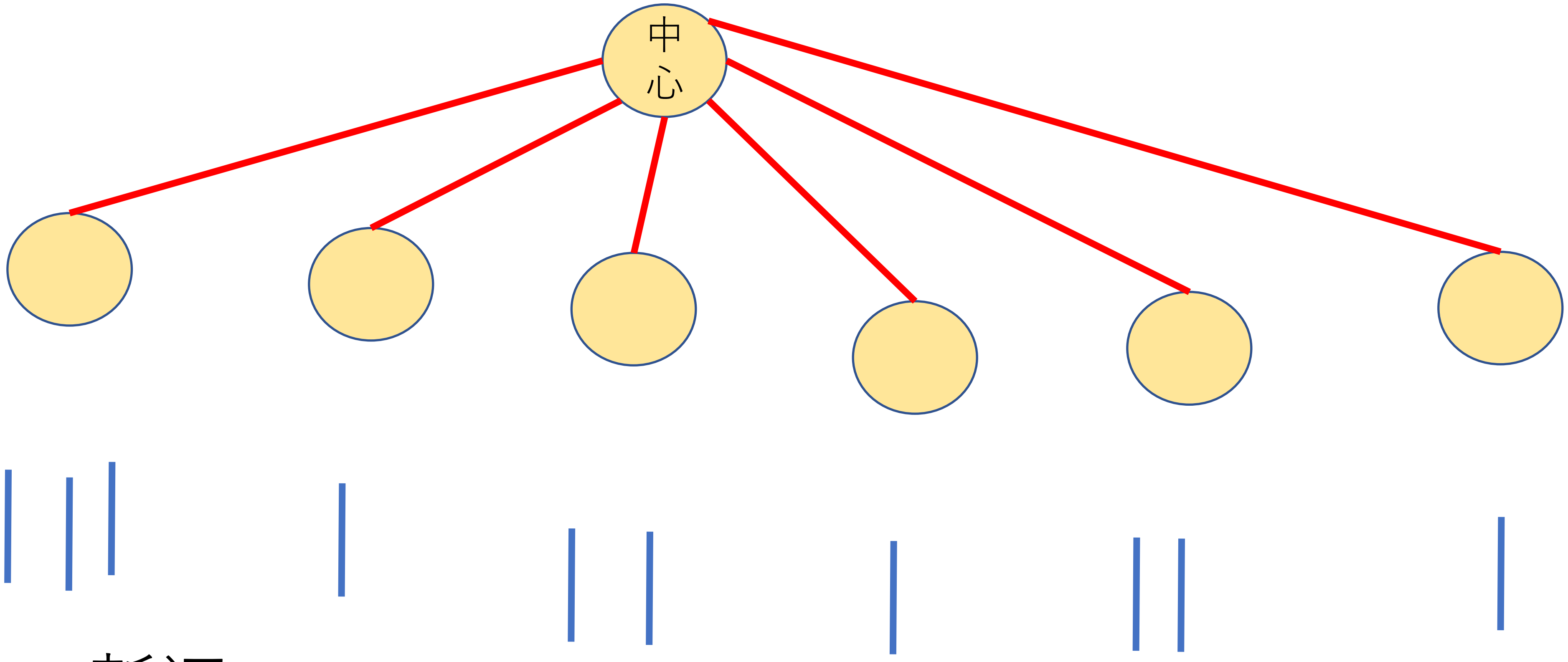
小課題7 (24点)追加の制約なし(N<=200000)

- 全体で各新辺を消したときのコストをpriority_queとかsetとかで管理
- 次の処理を繰り返す
- 全体から取り出しコスト最小の操作をする。つまり1本新辺を消して元辺を追加、連結成分をマージ
- 新たにできた連結成分について新辺を消したときのコストを求めて全体に入れる

小課題7 (24点)追加の制約なし(N<=200000)

- このマージのとき各連結成分はまだ使っていない元辺をもって、そのコスト最小値を求める必要がある（priqueueとかsetとかで管理）
- マージテクを使って辺の数が少ない方を大きい方に追加するようにするとO(Nlog^2N)で満点

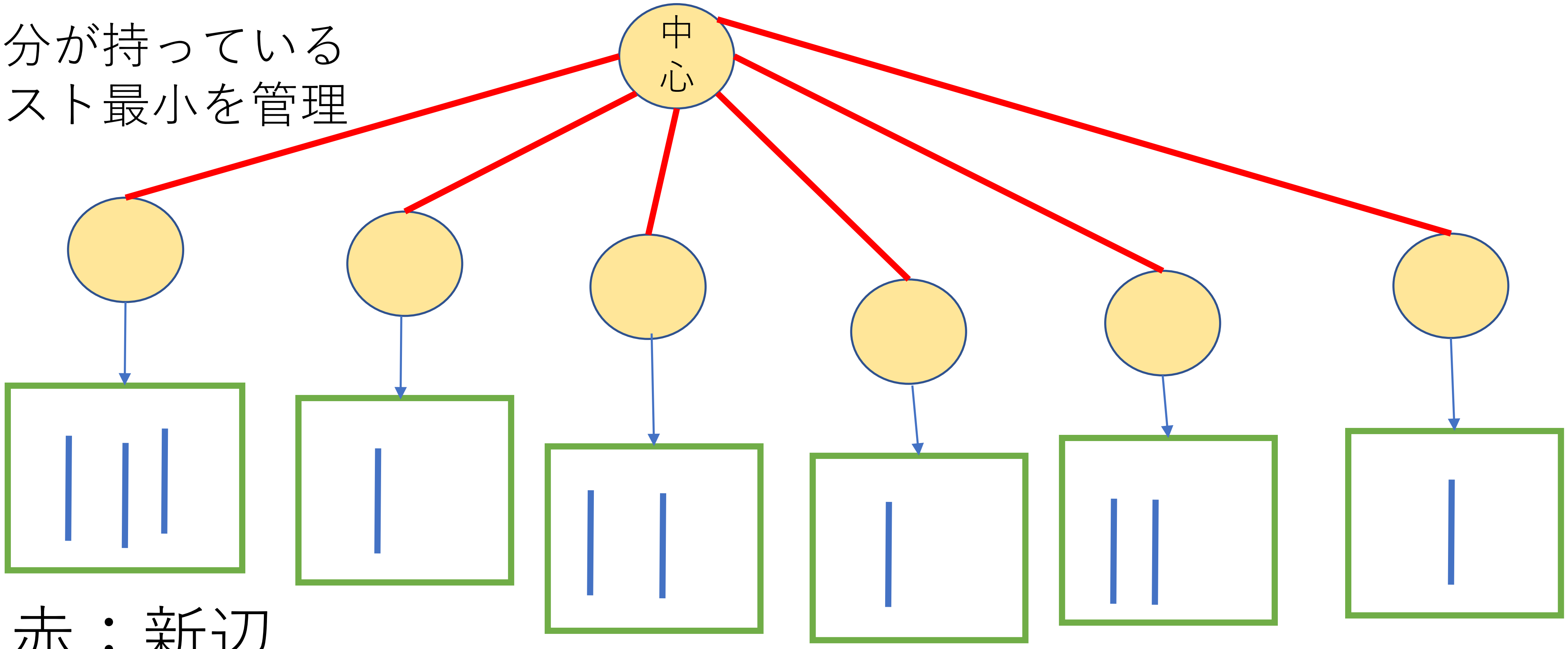
小課題7 (24点)追加の制約なし(N<=200000)



赤：新辺
青：元辺

小課題7 (24点)追加の制約なし(N<=200000)

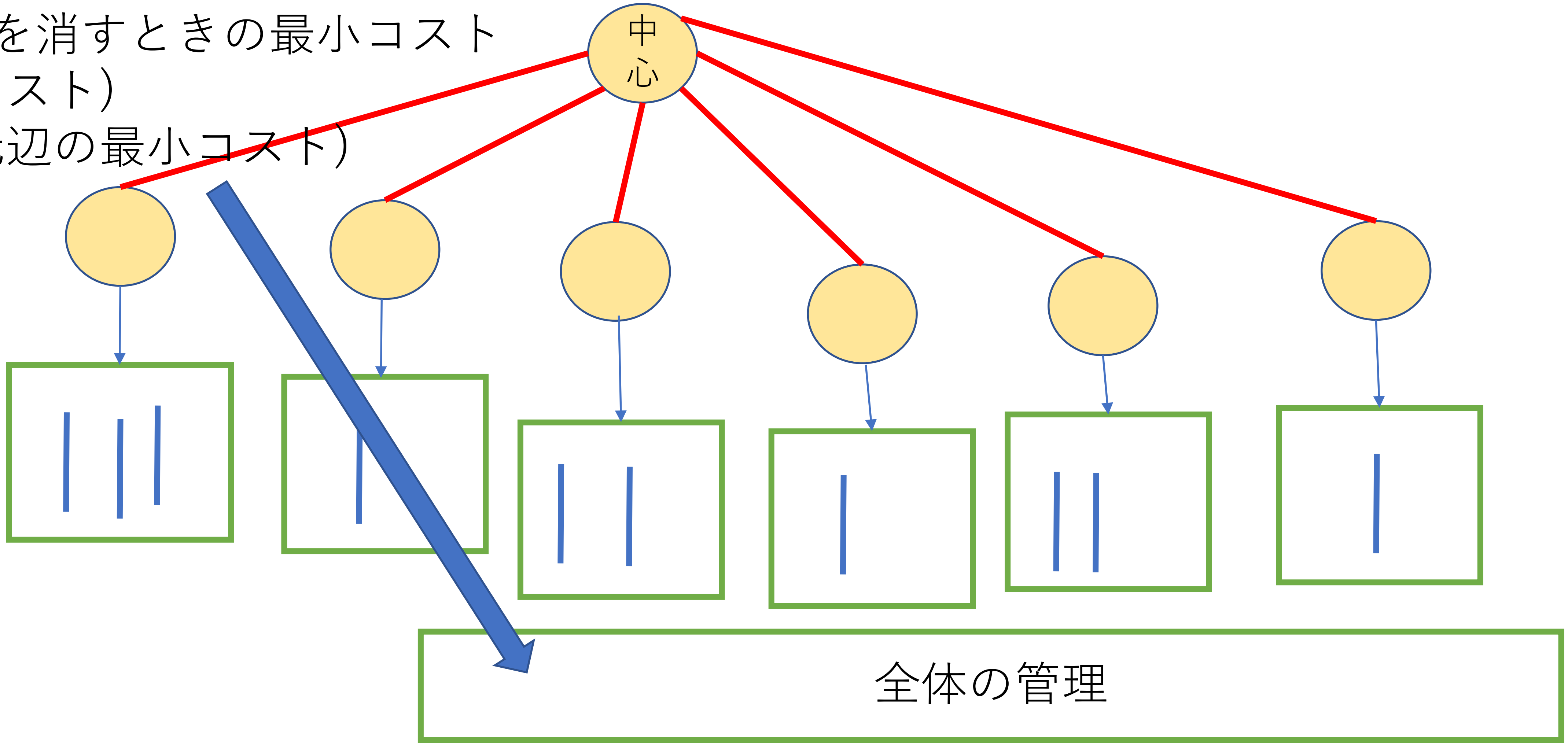
各連結成分が持っている
元辺のコスト最小を管理



赤：新辺
青：元辺

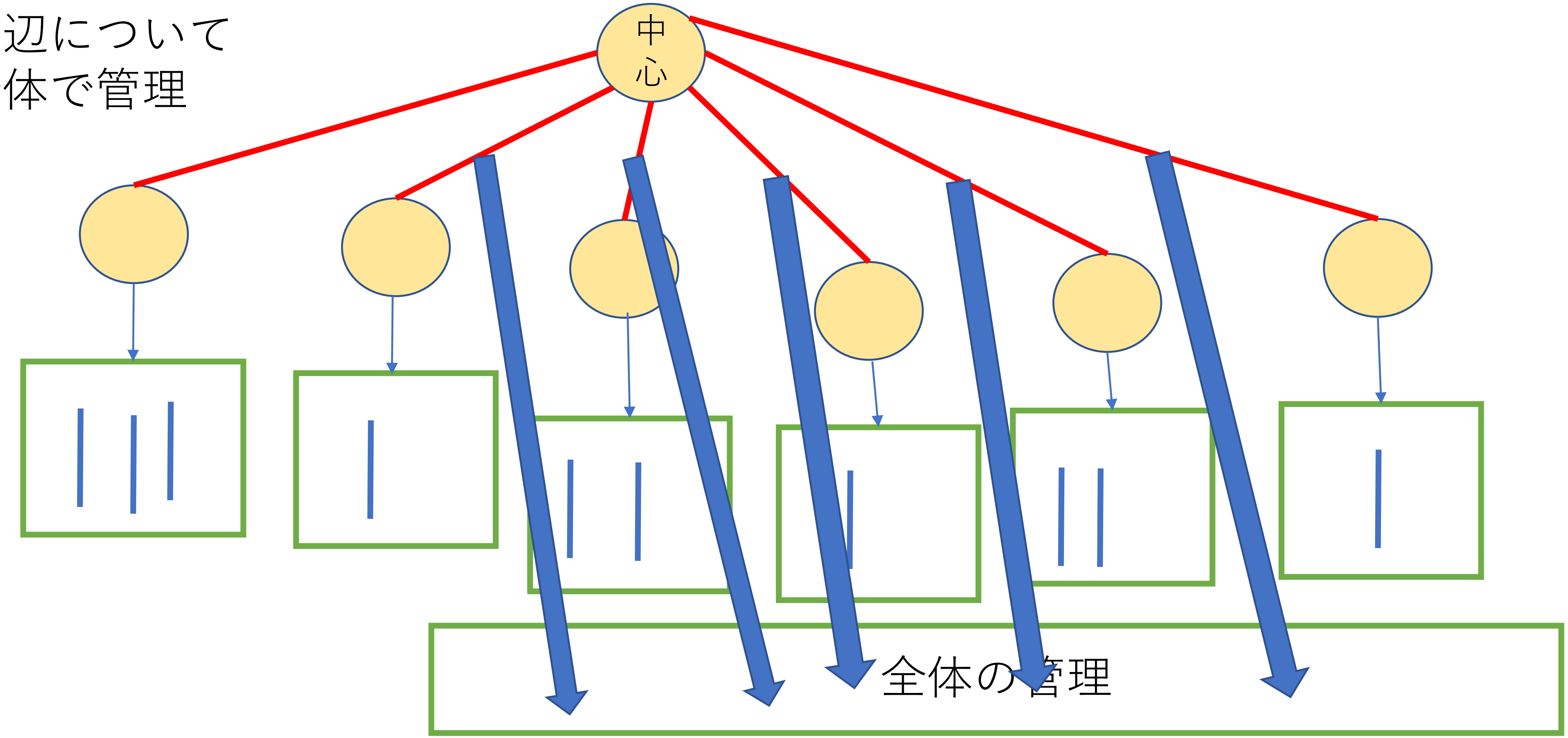
小課題7 (24点)追加の制約なし(N<=200000)

この赤辺を消すときの最小コスト
－(赤辺コスト)
＋(持つ元辺の最小コスト)
を入れる

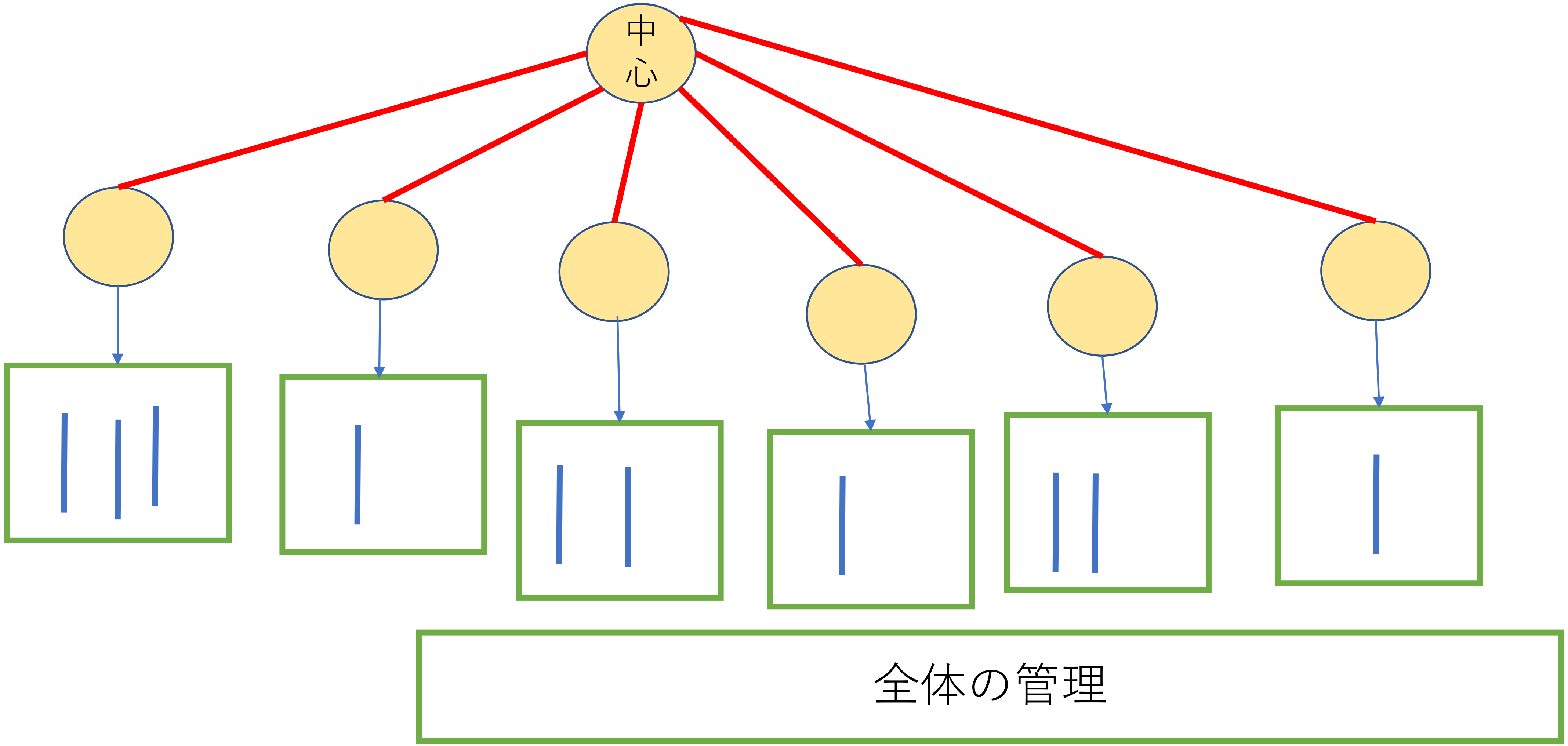


小課題7 (24点)追加の制約なし(N<=200000)

全部の赤辺について
入れて全体で管理

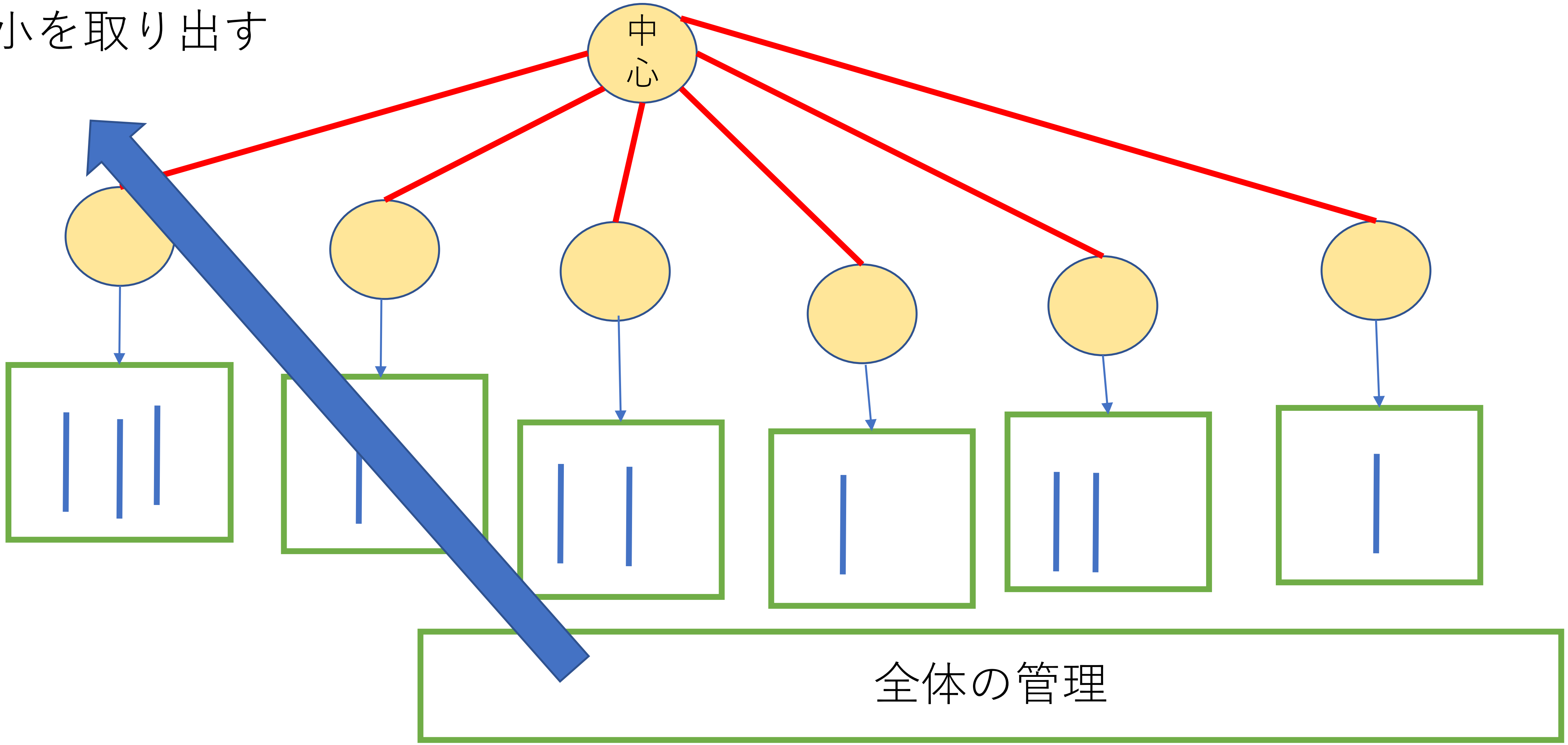


小課題7 (24点)追加の制約なし(N<=200000)



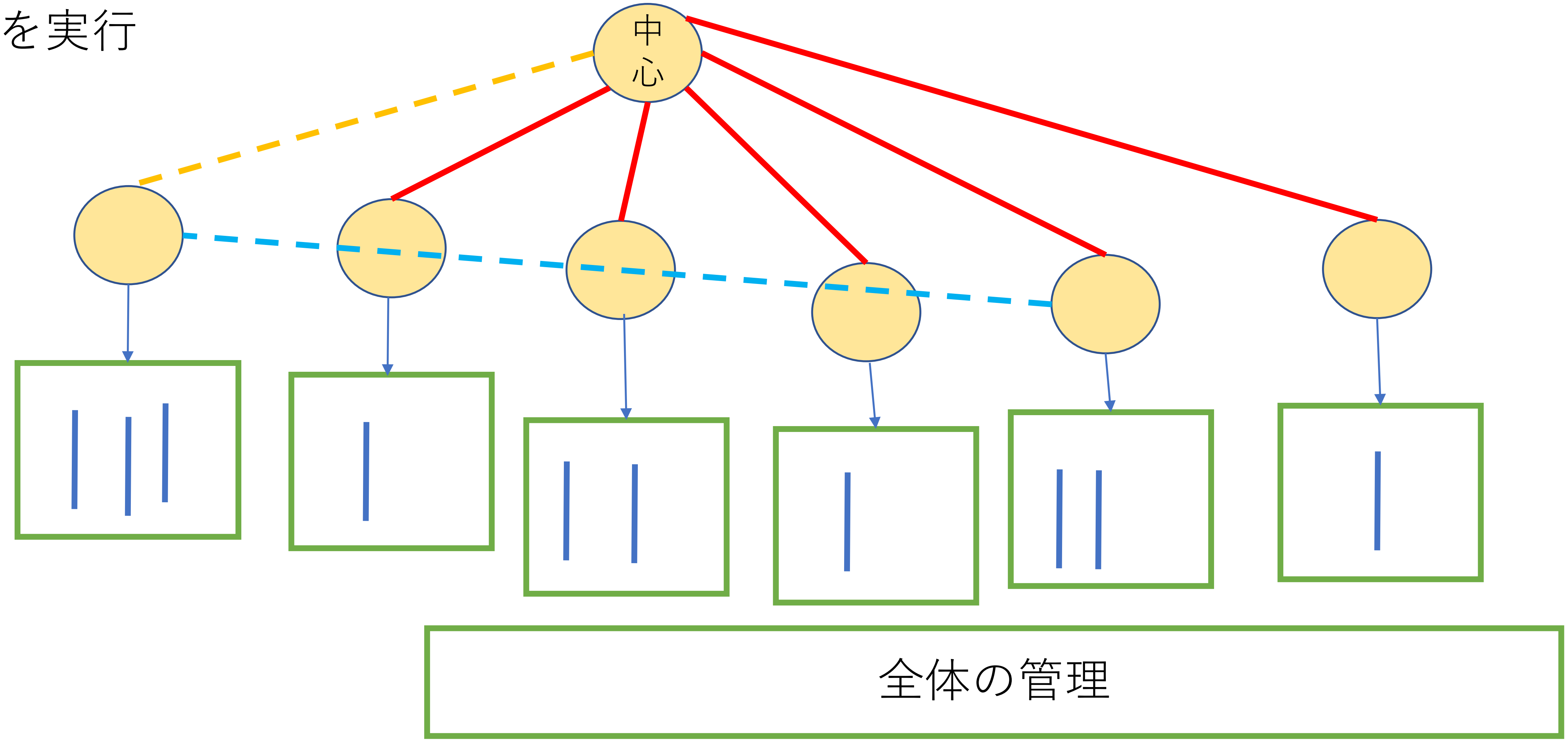
小課題7 (24点)追加の制約なし(N<=200000)

全体の最小を取り出す



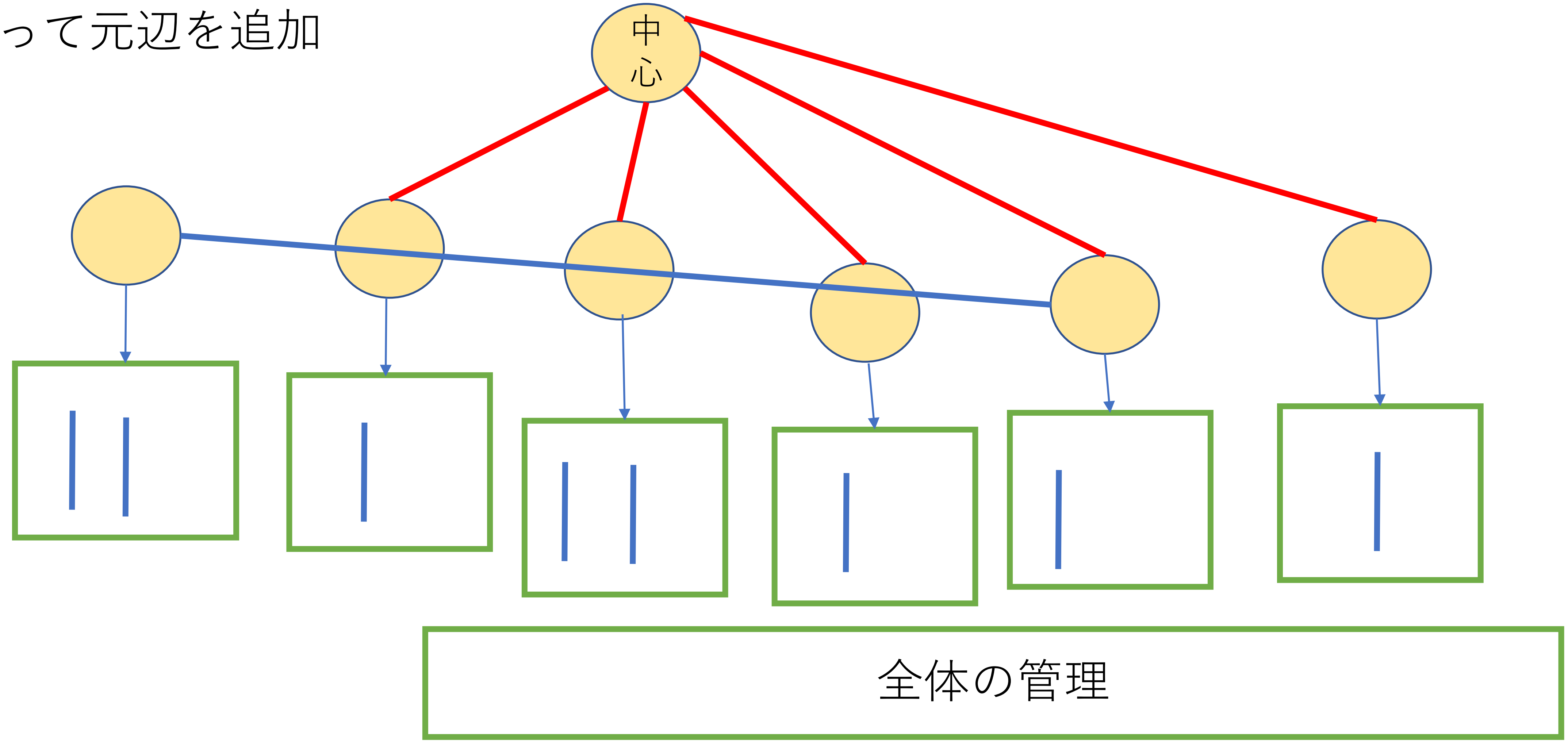
小課題7 (24点)追加の制約なし(N<=200000)

その操作を実行



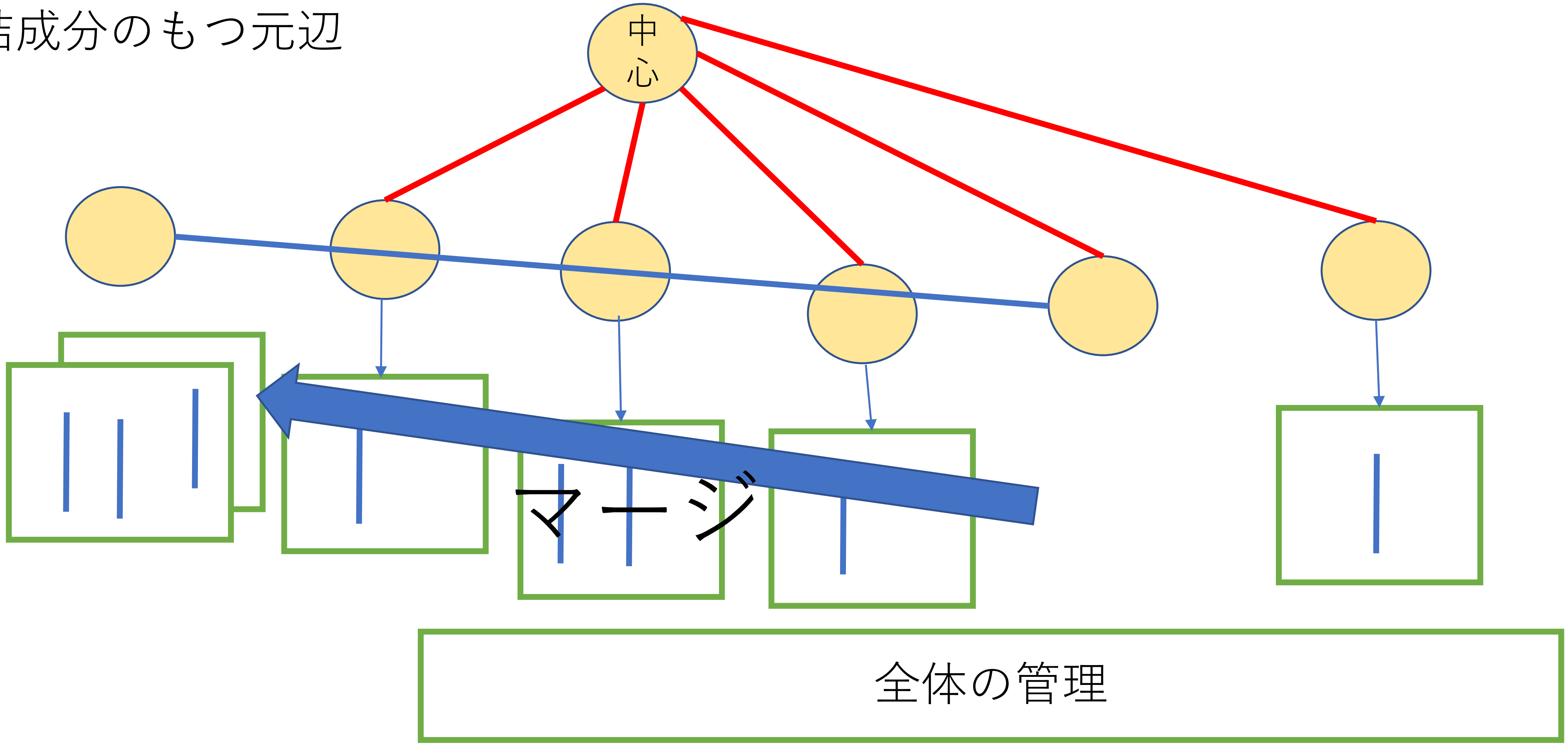
小課題7 (24点)追加の制約なし(N<=200000)

新辺を切って元辺を追加

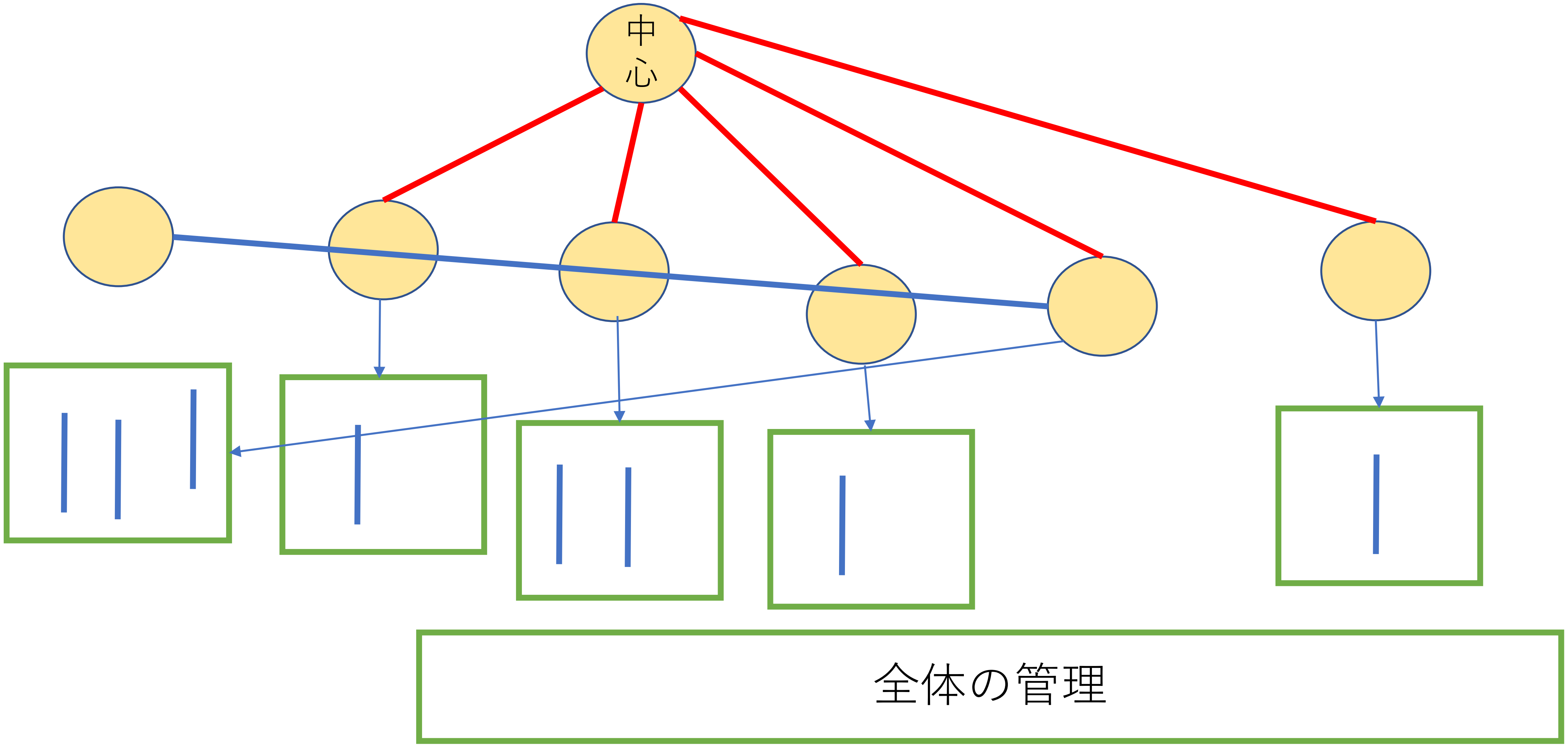


小課題7 (24点)追加の制約なし(N<=200000)

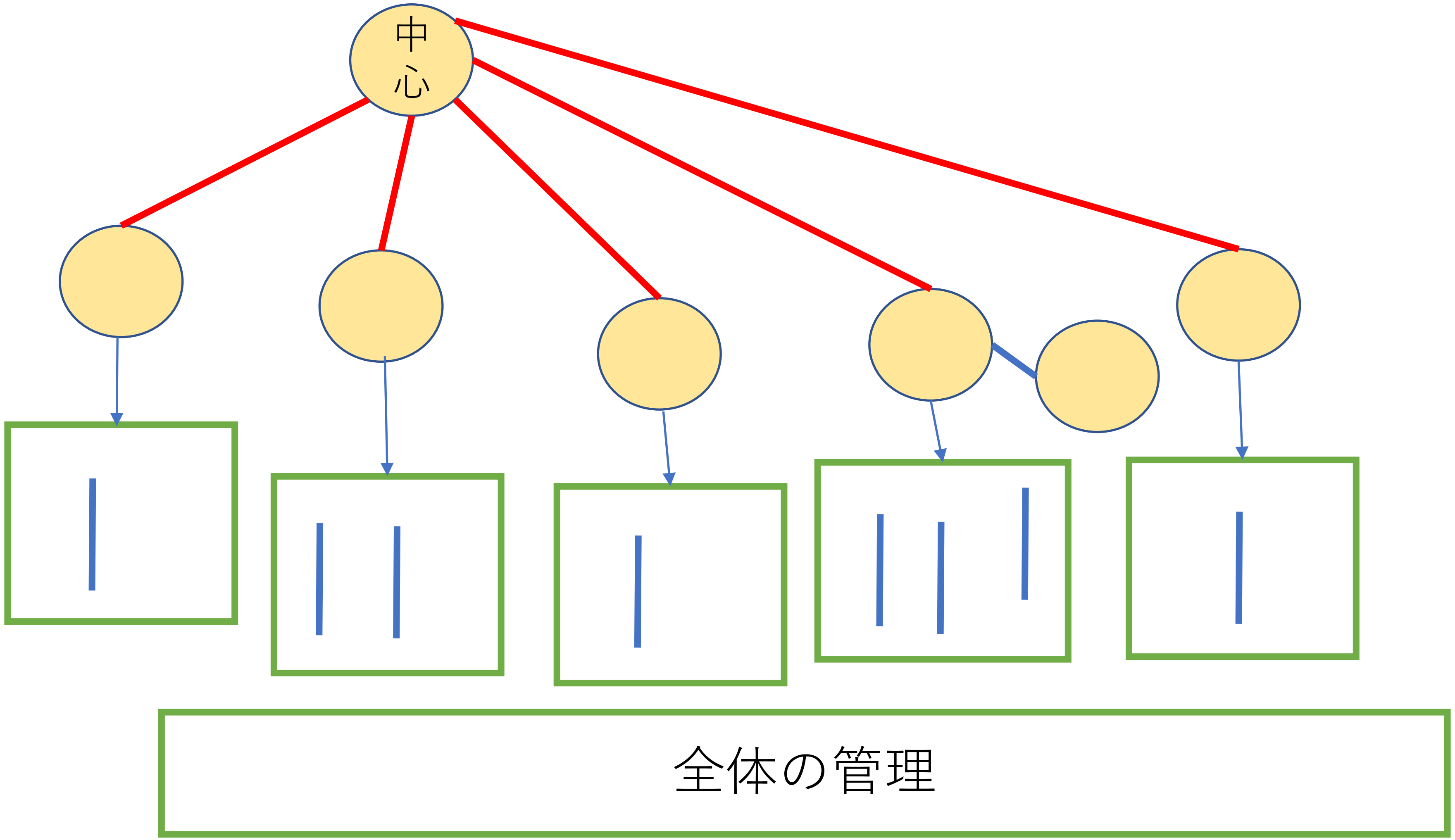
2つも連結成分のもつ元辺
をマージ



小課題7 (24点)追加の制約なし(N<=200000)

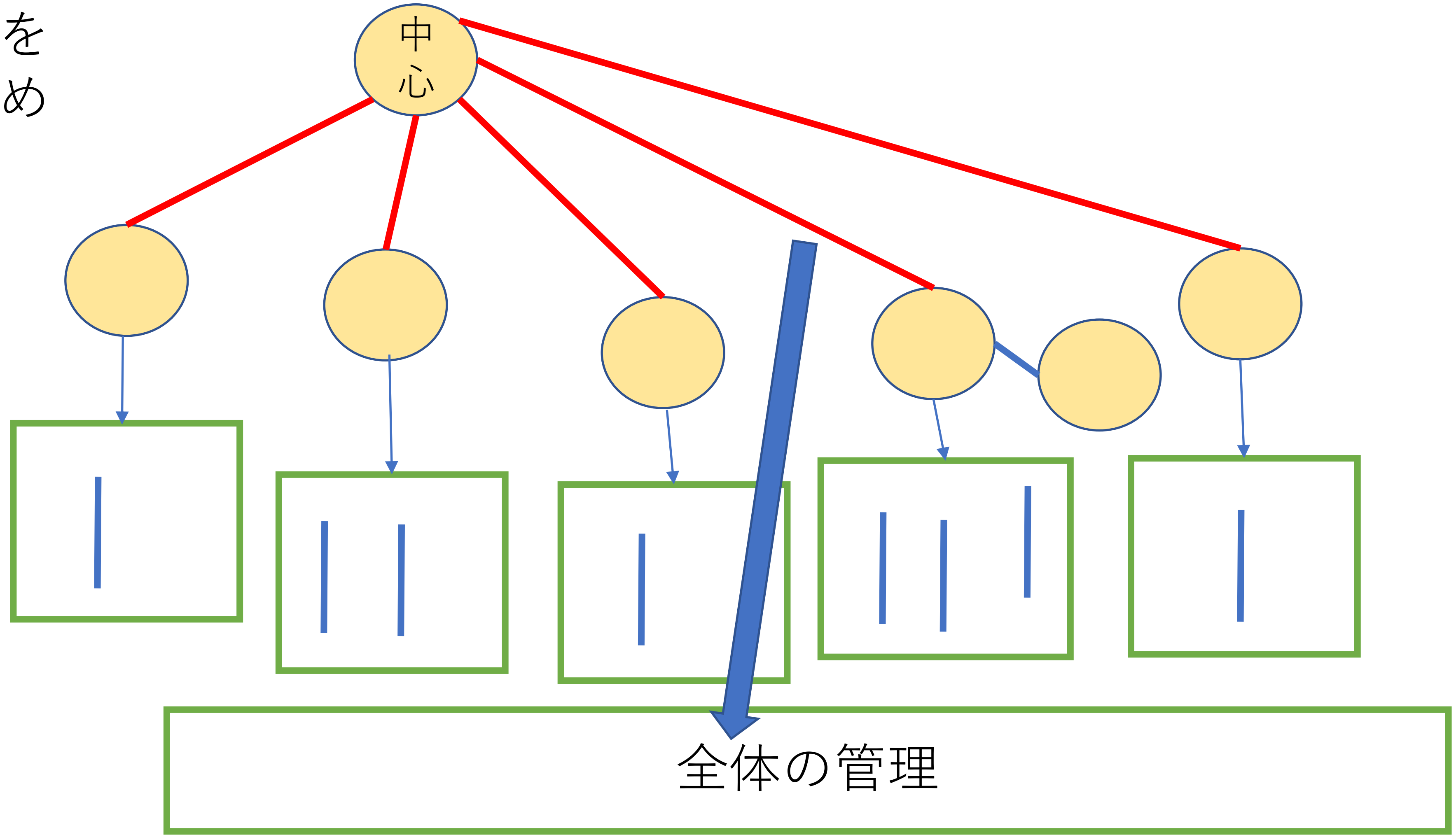


小課題7 (24点)追加の制約なし(N<=200000)



小課題7 (24点)追加の制約なし(N<=200000)

新しい連結成分の赤辺を消すときのコストを求めていれる



得点分布

- 76点　1人
- 58点　3人
- 50点　1人
- 45点　1人
- 37点　4人
- 25点　5人
- 12点　3人
- 0点　10人