環

情報Ⅰ

以下、法制度に関しては、日本のものについて考えるものとする。

(ア) 次の文章を読み、空欄 (1) ～ (5) にあてはまるものを選択肢から1つ選び、その番号を解答欄にマークしなさい。

二次創作につき特に問題となるのは (1) 性の判断、裁判例の表現によれば「表現上の本質的な特徴を直接感得」できるか (中略)、との点である。

(1) 性については、少なくとも原告作品の (2) 的表現が被告作品の表現に共通していること（ (2) 的表現の共通性）が必要となる。事実・ (3) ・ (2) 性のない表現が共通するに過ぎない場合には侵害が否定される (中略)。そして一般に、漫画等のキャラクターの名称・性格・設定等は抽象的な (3) と評価され、著作権法上保護されるのはキャラクターの具体的表現（漫画の絵等）に限られる (中略) と解されている。

つまり、二次創作作品が原作のキャラの (4) や設定、短い決め台詞、容姿の抽象的な特徴等を用いるに過ぎない場合 (中略)、原作の著作権・著作者人格権を侵害するものではない。このように二次創作作品の中には、原作の (5) （著作権法2条1項11号）ではなく、独立の新たな著作物と評価されるものが少なからず存在する。

（出典：金子敏哉「二次創作と著作権法」法学教室 449 号（2018 年）を一部改変）

【 (1) ～ (5) の選択肢】

(1) 創作	(2) 二次的著作物	(3) 新規	(4) 剽窃	(5) アイデア
(6) 名前	(7) 類似	(8) デザイン	(9) 共同著作物	(0) 進歩

(イ) 特許法に関する説明として、正しいものを次の選択肢から1つ選び、その番号を解答欄 (6) にマークしなさい。

(1) 特許権は、原則として設定登録から 70 年をもって消滅する。
(2) 発明の内容を、守秘義務を伴う契約に基づいて委託先の従業員に開示した場合、その発明は公知となるから、特許を受けることができない。
(3) 特許庁の審査基準は、人間を治療する方法を、特許権による保護の対象から除外している。
(4) 特許権者が適法に販売した特許製品は、特許権者の許諾なく転売することはできない。
(5) 自然法則を利用した技術的思想は、発明には該当しない。

(ウ) 著作権法に関する説明として、正しいものを次の選択肢から1つ選び、その番号を解答欄 (7) にマークしなさい。

(1) 授業の内容をそのまま手書きでノートに書き写す行為は、著作物の複製にはあたらない。
(2) 飲食店に設置したテレビにより、放送されているテレビ番組をその店の客に見せる行為は、著作権の侵害には該当しない。

(3) 著作物や題号に対し、著作者の意に反する変更、切除、その他の改変を加える行為は、同一性保持権の侵害を構成する。

(4) 著作権の存続期間は、原則として、著作物の創作から 20 年で満了する。

(5) 入学試験の問題として著作物を複製する場合には、事前に著作権者の許諾を得なければならない。

(エ) 個人情報の保護に関する法律（個人情報保護法）に関する説明として、正しいものを次の選択肢から 1 つ選び、その番号を解答欄 $\boxed{(8)}$ にマークしなさい。

(1) 氏名は、同姓同名の人が生存している場合に特定の個人を識別することができないから、個人情報には該当しない。

(2) 新聞に掲載された公知の情報は、個人情報には該当しない。

(3) 外国に居住する外国人の個人情報は、個人情報保護法による保護の対象には該当しない。

(4) 個人情報取扱事業者が、顧客との電話の通話内容を録音することは、書面での同意がない限り、個人情報保護法に違反する。

(5) 特定の政党が発行する新聞や機関誌等を購読しているという情報は、要配慮個人情報に該当しない。

2021年度 環境情報学部 一般選抜 問題訂正

教科・科目	ページ	設問	誤	→	正
数学 または 情報	15	[情報Ⅱ]	3行目～4行目 A(t)およびB(t)のグラフを描くと x=0からx=tの間のA(t)とB(t)で	→	A(x)およびB(x)のグラフを描くと x=0からx=tの間のA(x)とB(x)で
数学 または 情報	15	[情報Ⅱ]	7行目 $\int_0^t (A(u)-B(u))du=S$	→	$\int_0^t (A(x)-B(x))dx=S$

情報II

切符売り場や ATM などの窓口での処理とそのモデル化について述べている次の文章の空欄 $\boxed{(9)}.\boxed{(10)}$ 〜 $\boxed{(11)\,(12)}$ について、空欄に入るもっとも適切な数字をマークしなさい。また、空欄 $\boxed{(13)\,(14)}$ 〜 $\boxed{(25)\,(26)}$ に入るもっとも適した語を選択肢から選び、選択肢の番号を解答欄にマークしなさい。ただし、$\boxed{(13)\,(14)}$ と $\boxed{(15)\,(16)}$ は順不同である。

窓口が一つだとし、窓口でのサービスを受けるために人が次々とやってくる状況を考える。単位時間あたりにやってくる人の数の時間平均を到着率と呼び、λ とする。この場合、個々の人の到着する間隔は必ずしも等間隔ではないものとし、到着した人は、窓口がサービスを受けている人で埋まっていたら、必ず列に並んで待つものとする。また、窓口でサービスを受けている人 (0 人または 1 人) と列に並んでいる人 (0 人以上) の合計人数の時間平均を、平均系内人数と呼び、N とする。たとえば、1 時間のうち 30 分間は窓口でサービスを受けている人がいるだけで、残りの 30 分間は、待っている人が 1 人だった場合、N は $\boxed{(9)}.\boxed{(10)}$ 人になる。さらに、サービスを受けにやってきて、窓口あるいは列に並び始めてから窓口でのサービスを受け終わって離れるまでの時間の各利用者に関する平均を平均系内時間と呼び、T とする。たとえば、1 時間のうちに 3 人の到着があり、1 人は待ち時間 0 分で 20 分間窓口で対応を受け、残りの 2 人は、それぞれ 15 分待ってから窓口で 20 分間の対応を受けたとすると、T は $\boxed{(11)\,(12)}$ 分になる。

ある 1 人に着目した場合、その人が到着してから、窓口を離れるまでの間に、何人の人が到着するかを考えると、平均系内人数 N と平均系内時間 T と到着率 λ には、

$$\boxed{(13)\,(14)} = \boxed{(15)\,(16)}$$

という関係が成立していることが直感的に分かるが、これは次のように説明できる。

到着客は、列が長くても、あきらめることなく、並ぶものとし、時刻 0 から t の間に到着した人数の合計を、$A(t)$ とし、同じく、サービスを終えて系から立ち去った人数の合計を $B(t)$ とする。

このとき、常に

$$A(t)\ \boxed{(17)\,(18)}\ B(t)$$

が成立する。

n 番目に立ち去った客の系内時間を T_n とすると、時刻 t でサービス中の客の分も含んだ系内時間の総和は、たかだか、

$$\sum_{i=1}^{A(t)} T_i$$

である。また、時刻 t でサービス中の客の分を含まない系内時間の総和は、

$$\sum_{i=1}^{B(t)} T_i$$

になる。

また、時刻 t における系内人数は、$A(t)-B(t)$ なので、x 軸を時間とし、y 軸を人数として、$A(t)$ および $B(t)$ のグラフを描くと、$x=0$ から $x=t$ の間の $A(t)$ と $B(t)$ で囲われた領域の面積は、時刻 t でサービスを打ち切った場合の系内時間の総和 S になる。

別の表現をすると、

$$\int_0^t (A(u)-B(u))du = S$$

である。

この S に関して、次の不等式が成立する。

$$\boxed{(19)}\boxed{(20)} \leqq S \leqq \boxed{(21)}\boxed{(22)}$$

この式を t で割り算すると、中辺は、$\dfrac{S}{t}$ となるが、十分に大きな t に対して、これは、$\boxed{(23)}\boxed{(24)}$ を表している。

このとき、両辺を、$\dfrac{A(t)}{t}$ または、$\dfrac{B(t)}{t}$ が表れるように変形してやると、十分に t を大きくすると、$\dfrac{A(t)}{t}$ および $\dfrac{B(t)}{t}$ は、ともに $\boxed{(25)}\boxed{(26)}$ に近づく。したがって、左辺と右辺が同じ値に近づくことになり、等式が得られ、最初の式、

$$\boxed{(13)}\boxed{(14)} = \boxed{(15)}\boxed{(16)}$$

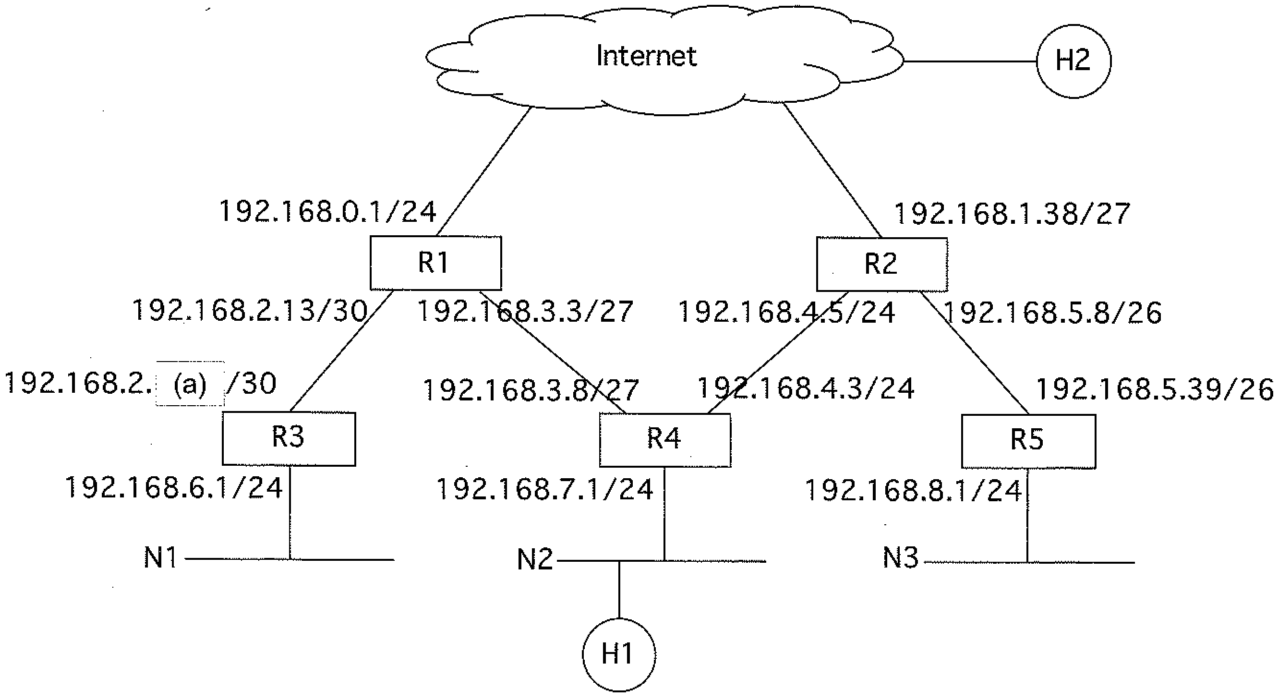
が得られる。

【$\boxed{(13)}\boxed{(14)}$～$\boxed{(25)}\boxed{(26)}$ の選択肢】

(11)	N	(12)	T	(13)	λ	(14)	NT
(15)	$N\lambda$	(16)	$T\lambda$	(17)	$\leqq$	(18)	$\geqq$
(19)	$=$	(20)	$\neq$	(21)	$\sum_{i=1}^{A(t)} T_i$	(22)	$\sum_{i=1}^{B(t)} T_i$

情報III

次のようなネットワークがある。



(ア) 図の空白 (a) に入るもっとも適切な数字を解答欄 (27)(28)(29) にマークしなさい。

(イ) ネットワーク N2 に接続されているコンピュータ H1 から H2 に向けてパケットを送信したところ、R4 → R1 →インターネットとパケットが中継されてインターネットに出ていった。また、192.168.1.38 にパケットを送信したところ、R4 → R2 とパケットが中継された。次に示す R4 の経路表の (30)(31)(32) ～ (38)(39)(40) に入るもっとも適切な数字を解答欄にマークしなさい。

ネットワーク	次ホップルータ
default	192.168. (30)(31)(32) . (33)(34)(35)
192.168.3.0/27	直接接続
192.168.4.0/ (36)(37)	直接接続
192.168.7.0/24	直接接続
192.168.1. (38)(39)(40) /27	192.168.4.5

(ウ) 次の文章の (41) ～ (55)(56)(57) に入るもっとも適切な数字を解答欄にマークしなさい。

上に示す R4 の経路表では、H1 からネットワーク N (41) には到達できないか、または遠回りすることとなる。その理由は H1 から N (41) に接続されているコンピュータに送信されたパケットは、R4 から

R $\boxed{(42)}$ に送られるためである。一方、ネットワーク N $\boxed{(43)}$ に送られるパケットは R $\boxed{(44)}$ と R $\boxed{(45)}$ の経路が正しく設定されていれば、遠回りすることなく到達することができる。

ネットワーク N $\boxed{(41)}$ に遠回りせずにパケットを届けるためには、R4 の経路表に下記のエントリを追加する必要がある。

ネットワーク	次ホップルータ
192.168. $\boxed{(46)}\boxed{(47)}\boxed{(48)}$. $\boxed{(49)}\boxed{(50)}\boxed{(51)}$ /24	192.168. $\boxed{(52)}\boxed{(53)}\boxed{(54)}$. $\boxed{(55)}\boxed{(56)}\boxed{(57)}$

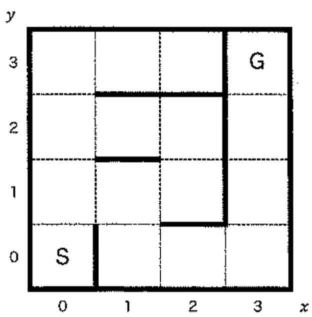
環

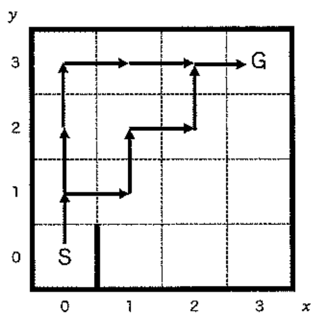
情報IV

自律型のロボットが $n \times n$ 区画の迷路を走行するマイクロマウスの探索アルゴリズムを考える。次の文章の空欄 $\boxed{(58)}\boxed{(59)}$ から $\boxed{(72)}\boxed{(73)}$ には適切な数字を解答欄にマークしなさい。また、空欄 $\boxed{(74)}\boxed{(75)}$ から $\boxed{(92)}\boxed{(93)}$ にはもっとも適したものを選択肢から選び、解答欄にマークしなさい。

(ア) 実際のマイクロマウス競技では、16×16（もしくは 32×32）区画の迷路中央にゴール地点が存在するため、迷路解析の代表的な手法である左手法等ではゴールまで到達することができない可能性がある。また、走行可能な持ち時間と回数も限られている。このような状況でも迷路の探索と最短経路の導出を同時に行える方法を考えてみよう。

簡単のため、次図（左）のような 4×4 の迷路を想定し、スタート座標 $S(0,0)$、ゴール座標 $G(3,3)$ とする。最初の状態ではスタート地点周囲以外の壁はどこにあるかわからないため、スタート地点周囲以外に壁は無いと仮定して最短経路を考えると、次図（右）のように、一歩進んだ後、ゴールに向かって斜めギザギザに進む、もしくは3マス直進し、右折後3マス直進する経路などが最短経路となる。





実際には予定していた経路上に壁があれば通れないため、その時点で把握している迷路情報から導き出されるゴールへの最短経路を進み、必要に応じて最短経路を再導出する、次のような手順をとることにしよう。

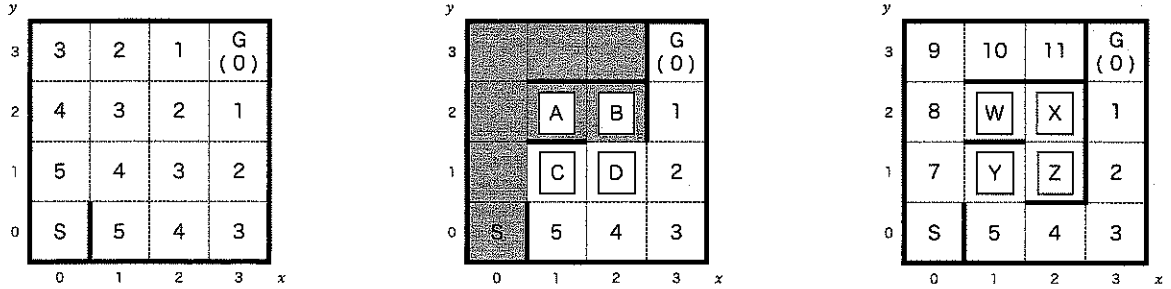
1. ゴールへの最短経路に沿って1マス進む
2. 壁を読み取る
3. 最短経路を導き直す

これをゴールに到達するまで繰り返せばよい。

ここで、ゴールの歩数を0とし、移動可能な隣接マスを1、その隣を2、$\cdots$ と、ゴールからの歩数をマップ化したものを歩数マップと呼ぼう。最初にマウスが認識している壁はスタート地点のもののみであり、この段階での歩数マップは次図（左）のようになる。

既知の壁情報をもとに歩数マップを作成し、順次現在のマスより歩数の小さいマスへ移動し、1マス移動するごとに壁情報と歩数マップを更新し、次に進む方向を決定する。グレーのマスを探索した時点

での歩数マップの一部は次図（中）のようになる。このときマス A、B、C、D に入る歩数は、それぞれ [(58)(59)]、[(60)(61)]、[(62)(63)]、[(64)(65)] となる。壁の探索が終わると次図（右）のような歩数マップが完成する。このときマス W、X、Y、Z に入る歩数は、それぞれ [(66)(67)]、[(68)(69)]、[(70)(71)]、[(72)(73)] となる。タイムアタックでは歩数マップにしたがって進めば最短経路でゴールに到達する。



(イ) 上記の方針に従い、8×8 区画の迷路を対象とした探索アルゴリズムを記述すると次のようになる。ここで、$M_{i,j}$ は座標 (i,j) の歩数を、$W_{i,j,n}$、$W_{i,j,e}$、$W_{i,j,s}$、$W_{i,j,w}$ はそれぞれ座標 (i,j) の北・東・南・西側の壁情報を保持する変数である。ただし、マップの上方向が北であり、壁情報は 0（壁なし）、1（壁あり）、2（未探索）のいずれかの値をとる。スタート位置は座標 $(0,0)$ とし、スタート位置東側の壁は既知であるとする。

```
北方向に 1 マス直進する
到達マスがゴールでなければ次の処理 A を繰り返す
処理 A の始まり
    壁の情報を取得・更新する
    歩数マップを更新する 【歩数マップ作成・更新アルゴリズム】
    前進方向マスの歩数マップの値が最小かつ移動可能であれば 1 マス前進する
    そうでなければ処理 B を実行する
    処理 B の始まり
        左隣接マスの歩数マップの値が最小かつ移動可能であれば左へ転回、1 マス進む
        そうでなければ処理 C を実行する
        処理 C の始まり
            右隣接マスの歩数マップの値が最小かつ移動可能であれば右へ転回、1 マス進む
            そうでなければ処理 D を実行する
            処理 D の始まり
                180 度転回し 1 マス戻る
            処理 D の終わり
        処理 C の終わり
    処理 B の終わり
    進行方向と座標を更新する
処理 A の終わり
```

【歩数マップ作成・更新アルゴリズム】

ゴール $M_{m,n}$ の値（歩数）を 0 とし、ゴール以外の全てのマス $M_{x,y}$ の値を未定義を意味する 255 とする

処理 A の始まり

　変数 f、変数 i の値をともに 0 にする

　条件 $i < 8$ が成立する間、処理 B を実行する

　処理 B の始まり

　　変数 j の値を 0 にする

　　条件 $\boxed{(74)}\boxed{(75)}$ が成立する間、処理 C を実行する

　　処理 C の始まり

　　　条件 $\boxed{(76)}\boxed{(77)}$ が成立する場合、処理 H に飛ぶ

　　　条件 $\boxed{(78)}\boxed{(79)}$ が成立する場合、処理 D を実行する

　　　処理 D の始まり

　　　　$M_{i,j+1}$ の値が 255 である場合、$M_{i,j+1}$ の値を $\boxed{(80)}\boxed{(81)}$ にする

　　　　変数 f の値を 1 にする

　　　処理 D の終わり

　　　条件 $\boxed{(82)}\boxed{(83)}$ が成立する場合、処理 E を実行する

　　　処理 E の始まり

　　　　$M_{i+1,j}$ の値が 255 である場合、$M_{i+1,j}$ の値を $\boxed{(84)}\boxed{(85)}$ にする

　　　　変数 f の値を 1 にする

　　　処理 E の終わり

　　　条件 $\boxed{(86)}\boxed{(87)}$ が成立する場合、処理 F を実行する

　　　処理 F の始まり

　　　　$M_{i,j-1}$ の値が 255 である場合、$M_{i,j-1}$ の値を $\boxed{(88)}\boxed{(89)}$ にする

　　　　変数 f の値を 1 にする

　　　処理 F の終わり

　　　条件 $\boxed{(90)}\boxed{(91)}$ が成立する場合、処理 G を実行する

　　　処理 G の始まり

　　　　$M_{i-1,j}$ の値が 255 である場合、$M_{i-1,j}$ の値を $\boxed{(92)}\boxed{(93)}$ にする

　　　　変数 f の値を 1 にする

　　　処理 G の終わり

　　　処理 H: 変数 j の値に 1 を加える

　　処理 C の終わり

　　変数 i の値に 1 を加える

　処理 B の終わり

　変数 f が 1 であれば処理 A の先頭に飛ぶ

処理 A の終わり

環

【$\boxed{(74)}\boxed{(75)}$〜$\boxed{(92)}\boxed{(93)}$ の選択肢】

(11) $j < 8$	(12) $i < 8$	(13) $j < i$	(14) $i < j$

(15) $M_{i,j}$ の値が 255 である　(16) $M_{i,j}$ の値が 0 である

(17) $W_{i,j,n}$ の値が 1 である　(18) $W_{i,j,n}$ の値が 0 または 2 である

(19) $W_{i,j,e}$ の値が 1 である　(20) $W_{i,j,e}$ の値が 0 または 2 である

(21) $W_{i,j,s}$ の値が 1 である　(22) $W_{i,j,s}$ の値が 0 または 2 である

(23) $W_{i,j,w}$ の値が 1 である　(24) $W_{i,j,w}$ の値が 0 または 2 である

(25) $M_{i,j}$	(26) $M_{i,j}+1$	(27) $M_{i,j}-1$	(28) $M_{i+1,j}$
(29) $M_{i+1,j}+1$	(30) $M_{i+1,j}-1$	(31) $M_{i,j+1}$	(32) $M_{i,j+1}+1$
(33) $M_{i,j+1}-1$	(34) $M_{i-1,j}$	(35) $M_{i-1,j}+1$	(36) $M_{i-1,j}-1$
(37) $M_{i,j-1}$	(38) $M_{i,j-1}+1$	(39) $M_{i,j-1}-1$	

環

情報V

空欄に入るもっとも適切なものを下の選択肢から選び、その番号をマークしなさい。

スポーツ大会などで行われるトーナメントについて考える。トーナメント図とは、図1のようにチーム名を一列に並べ、コの字型の線で対戦する2チームを結んだものである。図1の例では、チームAとチームBが試合 M_1 で対戦、チームCとチームDが試合 M_2 で対戦、さらに M_1 の勝者と M_2 の勝者が試合 M_3 で対戦することを表している。チーム名が入る箇所をスロットと呼び、端から順に0から始まる番号を付ける。また、ある試合 M に対して、M まで勝ち上がってくる可能性のあるスロット番号の集合を $P(M)$ とする。図1では $P(M_1) = \{0,1\}$, $P(M_3) = \{0,1,2,3\}$ である。

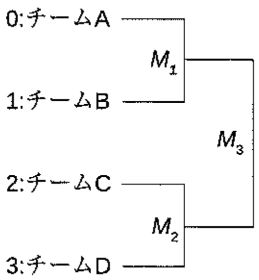


図1

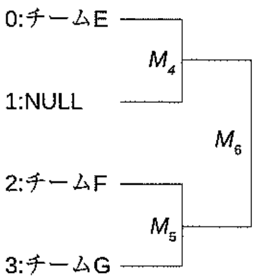


図2

ここでは、次のようなトーナメント図を作ることにする。

参加チーム数を n とすると、$n = 2^h$ となる整数 h がある場合は、すべてのチームが h 回の試合に勝つと優勝となるようなトーナメント図を作る。なお、対戦を表す線は交差しないように作るものとする。この場合、どの試合においても対戦する2チームは同じ回数の試合に勝っているため、最初の試合が1回戦、1回戦の勝者同士の対戦が2回戦、というように、ある試合が何回戦であるかを不整合なく定義できる。

$n = 2^h$ となる整数 h がない場合は、$2^{g-1} < n < 2^g$ を満たす整数 g に対して $2^g - n$ 個のNULLという仮想的なチームを追加して、チーム数が 2^g のトーナメント図を上で述べたように作る。そして、NULLと対戦する試合を不戦勝と考えることにする。例えば、図2ではチームEの最初の試合が不戦勝となる。なお、NULL同士の対戦がある場合は、その試合の勝者をNULLとして、2回戦以降の対戦でも同様にNULLと対戦するチームを不戦勝とする。

どのスロットにNULLを配置するかについては、トーナメント図のどの部分をとってもなるべく同じ割合でNULLが含まれるようにしたい。これを正確に定義するため、試合 M に対して $f(M)$ を、$P(M)$ の中に配置されるNULLの個数とする。例えば、図2では $f(M_4) = 1$、$f(M_5) = 0$、$f(M_6) = 1$ となる。あるトーナメント図が均等な配置であるとは、2回戦以降の任意の試合 M に対して、M が M' の勝者と M'' の勝者の対戦であるならば、$|f(M') - f(M'')| \leqq 1$ が成り立つことであると定義する。

次のアルゴリズムは、チーム数 n が与えられた時に、均等な配置になるようNULLを配置するスロッ

ト番号を出力するアルゴリズムである。均等な配置は複数あり得るが、その中の 1 つを出力するものとなっている。ただし、$\lceil x \rceil$ は x より大きいか等しい整数のうち最小のもの、$\lfloor x \rfloor$ は x より小さいか等しい整数のうち最大のもの、$x \bmod y$ は x を y で割った余りを表すものとする。

> 変数 n の値を与えられたチーム数とする。
> 変数 g の値を $\boxed{(94)}\boxed{(95)}$ とする。
> 変数 i の値を最初 0 とし、1 ずつ増やしながら $2^g - 1$ まで処理 A を繰り返す。
> 処理 A の始め
> 　変数 a の値を i とする。
> 　変数 b の値を 0 とする。
> 　処理 B を g 回繰り返す。
> 　処理 B の始め
> 　　変数 b の値を $2b + (\boxed{(96)}\boxed{(97)})$ とする。
> 　　変数 a の値を $\boxed{(98)}\boxed{(99)}$ とする。
> 　処理 B の終わり
> 　もし $b < \boxed{(100)}\boxed{(101)}$ なら i の値を出力する。（命令 C）
> 処理 A の終わり

上のアルゴリズムが正しいことは次のようにして証明できる。

まず、$0 \leqq x < 2^g$ である整数 x に対して、x を 2 進法 g 桁（上位の桁が 0 の場合も含む）で表現して各桁の数字を逆順に並べた数を $r_g(x)$ と定義する。つまり、$x = x_{g-1}2^{g-1} + x_{g-2}2^{g-2} + \ldots + x_1 2^1 + x_0 2^0$ だとすると、$r_g(x) = x_0 2^{g-1} + x_1 2^{g-2} + \ldots + x_{g-2}2^1 + x_{g-1}2^0$ である。ただし $x_0, \ldots, x_{g-1}$ は 0 または 1 とする。また、整数の集合 X のすべての要素 x が $0 \leqq x < 2^g$ を満たす時、$r_g(X) = \{r_g(x) \mid x \in X\}$ と定義する。

補題 1: 命令 C を実行する時、$b = r_g(i)$ という関係が成り立っている。

補題 1 の証明：

処理 A の実行を開始する時点での i の値を $i_{g-1}2^{g-1} + i_{g-2}2^{g-2} + \ldots + i_1 2^1 + i_0 2^0$（ただし $i_0, \ldots, i_{g-1}$ は 0 または 1）とする。処理 B の 1 回目の実行を終了した時点では、$a = i_{g-1}2^{g-2} + i_{g-2}2^{g-3} + \ldots + i_1 2^0$ かつ $b = \boxed{(102)}\boxed{(103)}$ となる。2 回目の実行を終了した時点では、$a = i_{g-1}2^{g-3} + i_{g-2}2^{g-4} + \ldots + i_2 2^0$ かつ $b = \boxed{(104)}\boxed{(105)}$ となる。同様に g 回目の実行まで考えると、g 回目の実行を終了した時点では $b = i_0 2^{g-1} + i_1 2^{g-2} + \ldots + i_{g-2}2^1 + i_{g-1}2^0 = r_g(i)$ となる。

補題 2: 整数 $g > 0$ に対して $S = \{s \mid 0 \leqq s < 2^g, s \text{ は整数}\}$ とすると、$S = r_g(S)$ が成り立つ。

補題 2 の証明：

まず $s' \in r_g(S)$ とする。s' は r_g の定義から 2 進法で表した時に g 桁（上位の桁が 0 の場合も含む）なので $0 \leqq s' < 2^g$ である。したがって $s' \in S$ となるから $S \supseteq r_g(S)$ である。

次に $s'' \in S$ とする。r_g の定義から $s'' = \boxed{(106)}\boxed{(107)}$ である。上で述べたことから $r_g(s'') \in S$ であるので

$s'' \in r_g(S)$ となり、したがって $S \subseteqq r_g(S)$ である。

上記2つのことから $S = r_g(S)$ である。

補題 3: スロットが 2^g 個あるトーナメント図と、任意の整数 m $(0 \leqq m \leqq 2^g)$ に対して、$r_g(s) < m$ となるスロット番号 s に NULL を配置すると、均等な配置になる。

補題 3 の証明：

g に関する帰納法で証明する。

$g = 1$ の場合、2 回戦以降の試合がないので明らかに成り立つ。

$g = k - 1$ の場合に成り立つと仮定し、$g = k$ の場合を考える。

まず、k 回戦（決勝戦）が均等な配置の条件を満たすことを示す。対戦するのが M' の勝者と M'' の勝者であり、$P(M') = \{0, 1, \ldots, 2^{k-1} - 1\}$、$P(M'') = \{2^{k-1}, 2^{k-1} + 1, \ldots, 2^k - 1\}$ であるとする。2 進法 k 桁で表現すると、$P(M')$ の要素は最上位桁が 0、$P(M'')$ の要素は最上位桁が 1 となる。したがって r_k の定義から $r_k(P(M'))$ の要素は $\boxed{(108)}\boxed{(109)}$、$r_k(P(M''))$ の要素は $\boxed{(110)}\boxed{(111)}$ となる。補題 2 から $r_k(P(M') \cup P(M''))$ には 0 から $2^k - 1$ までのすべての数が含まれるので、$r_k(s)$ が小さい s から順に NULL を配置していくと $P(M')$ と $P(M'')$ に交互に配置することになる。m が偶数の時は $f(M') = f(M'') = \boxed{(112)}\boxed{(113)}$ となり、m が奇数の時は $f(M') = \boxed{(114)}\boxed{(115)}$, $f(M'') = \boxed{(116)}\boxed{(117)}$ となるので、$|f(M') - f(M'')| \leqq 1$ が成り立つ。

次に、帰納法の仮定を利用するため、全体を $g = k$ のトーナメント図と考えて NULL を配置するスロットが、$P(M')$ と $P(M'')$ をそれぞれ $g = k - 1$ のトーナメント図と考えて NULL を配置するスロットと同じであることを示す。

m が奇数の場合の $P(M')$ の部分をまず考える。$P(M')$ の部分には $\boxed{(114)}\boxed{(115)}$ 個の NULL が配置されるので、集合 $T = \{s \mid 0 \leqq s < 2^{k-1},\ r_k(s) < m\}$ が集合 $T' = \{s \mid 0 \leqq s < 2^{k-1},\ r_{k-1}(s) < \boxed{(114)}\boxed{(115)}\}$ と同じであることを示せばよい。T の要素 s が $0 \leqq s < 2^{k-1}$ ということは、s の 2 進法 k 桁の表現から最上位の 0 を削ると s の 2 進法 $k-1$ 桁の表現になる。したがって $r_{k-1}(s) = \boxed{(118)}\boxed{(119)}$ であり、$r_k(s) < m$ という条件は $r_{k-1}(s) < \dfrac{m}{2}$ と同値であるが、m が奇数なのでこれは $r_{k-1}(s) < \boxed{(114)}\boxed{(115)}$ とも同値である。ゆえに $T = T'$ となる。

m が偶数の場合の $P(M')$ の部分についても同様である。また、$P(M'')$ の部分については、$g = k - 1$ のトーナメント図と考えた時のスロット番号は 2^{k-1} を引いたものになっていることに注意すれば同様に考えることができる。

したがって帰納法の仮定を利用することができ、$g = k$ の場合に配置した NULL は $P(M')$ と $P(M'')$ の部分でそれぞれ均等な配置になっている。k 回戦が均等な配置の条件を満たすことも最初に示したので、全体として均等な配置になっていることがわかる。

アルゴリズムの証明:

処理 A は、i の値を順に $0, 1, \ldots, 2^g - 2, 2^g - 1$ として繰り返すので、補題 1 から命令 C で出力される数の集合は $\{i \mid 0 \leqq i < 2^g,\ r_g(i) < \boxed{(100)}\boxed{(101)}\}$ となる。補題 3 から、この番号のスロットに NULL を

配置すると、均等な配置になる。

【$\boxed{(94)}\boxed{(95)}$ ～ $\boxed{(118)}\boxed{(119)}$ の選択肢】

(11) $\lceil \log_2 n \rceil$　(12) $\lfloor \log_2 n \rfloor$　(13) $\lceil \frac{a}{2} \rceil$　(14) $\lfloor \frac{a}{2} \rfloor$　(15) $a+b$

(16) $a-b$　(17) $a \bmod 2$　(18) $b \bmod 2$　(19) g　(20) n

(21) 2^g　(22) 2^g-n　(23) $\frac{m-1}{2}$　(24) $\frac{m}{2}$　(25) $\frac{m+1}{2}$

(26) $i_0 2^0$　(27) $i_0 2^1 + i_1 2^0$　(28) $i_0 2^{g-1}$　(29) $i_0 2^{g-1} + i_1 2^{g-2}$　(30) $2r_k(s)$

(31) $\frac{r_k(s)}{2}$　(32) $r_g(s'')$　(33) $r_g(r_g(s''))$　(34) 奇数　(35) 偶数

(36) 2^{k-1} 未満　(37) 2^{k-1} 以上