UFR de SCIENCES et TECHNIQUES
PAU

DATE/LIEU	EXAMEN:		DIPLÔME ET FILIERE	
	S5	**2017**	**L3**	**INFORMATIQUE**
	MODULE		COMPOSITION DE	
03/01/2017 **14h-16** **Amphi B**	**GRAPHE -TLI035CT**		**Examen**	
	Durée : **2h**		Nom de l'enseignant : K. **OURIACHI**	

Documents non autorisés

I-Optimisation du cycle d'assemblage d'un smartphone (8 points)

Pour améliorer la qualité de production et augmenter la fiabilité de ses smartphones, Samsung expérimente pour ses smartphones de la série A5, une automatisation intégrale du cycle de montage dans un atelier robotisé. Dans ce cycle montage on recense 12 tâches codées T1, T2,T3,T4,T5,T6, T7,T8, T9, T10, T11, et T12.La durée d'exécution de ces tâches varie dans un l'intervalle de faible amplitude noté [T_{min} ; T_{max}].

Pour chacune de ces tâches, le tableau ci-dessous indique la liste des robots qu'elle préempte (accès prioritaire sans partage) durant son exécution.

Tâche	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**
Liste de robots préemptés	**R1 R2 R3 R4**	**R1 R2 R4 R5 R7 R8**	**R2 R3 R7 R9**	**R2 R8 R9 R10**	**R1 R5 R6 R7 R9 R10**	**R2 R7 R8 R9**	**R2 R5 R9**	**R5 R10**	**R4 R6 R7**	**R1 R6 R9**	**R9 R10**	**R3 R5**

Deux problèmes sont soulevés par l'ingénieur en productique

- exploiter au maximum les possibilités de «paralléliser» l'exécution des tâches dans le but d'optimiser le cycle de montage,
- déterminer de façon **optimale** la **durée minimum** et la **durée maximum** pour ce cycle.

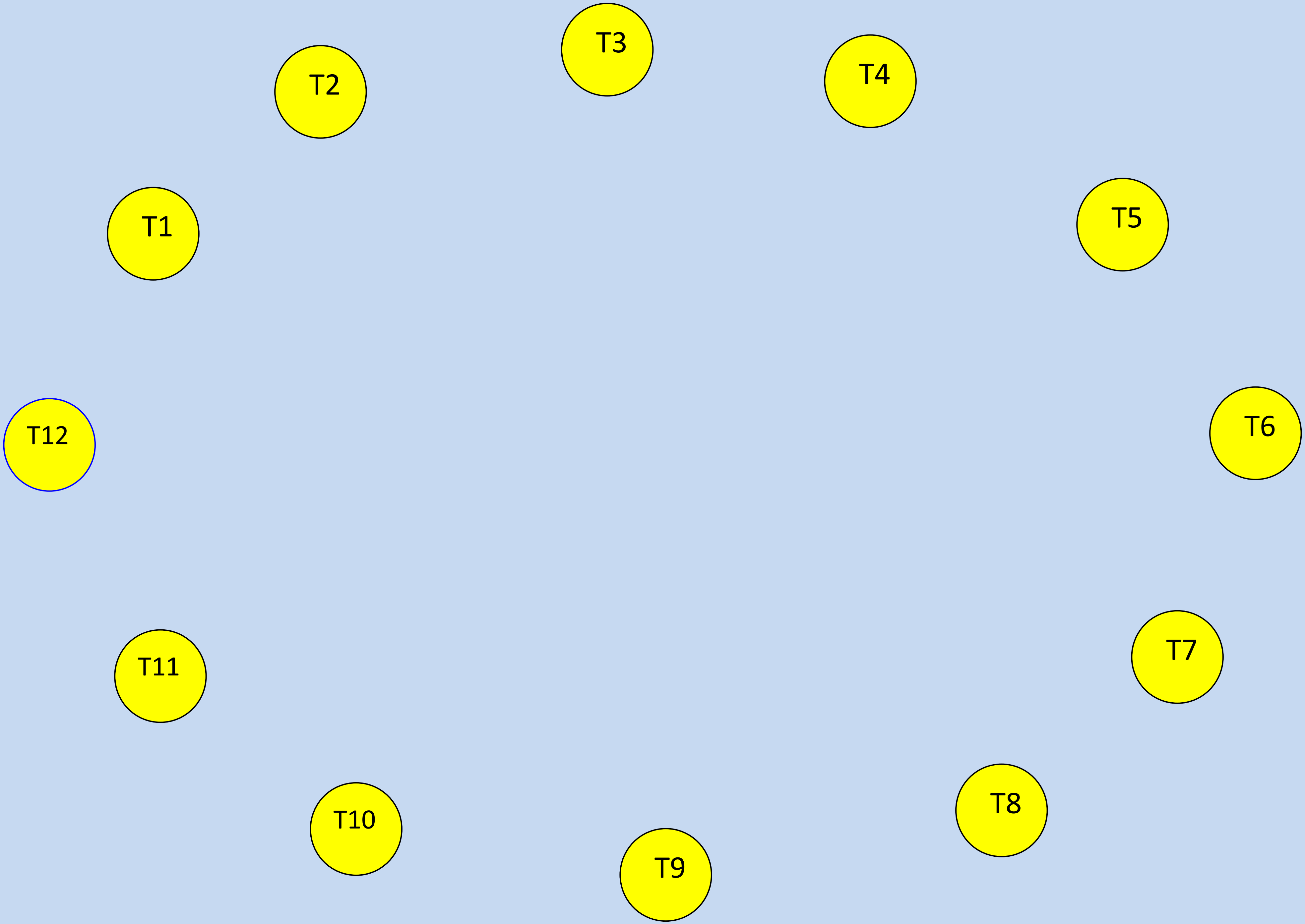
L'ingénieur logiciel saisi du problème doit proposer des **solutions systématiques** Dans cet objectif, il procède en 3 étapes :

1-Commencer par proposer un modèle de graphe qui met en évidence l'incompatibilité des tâches.

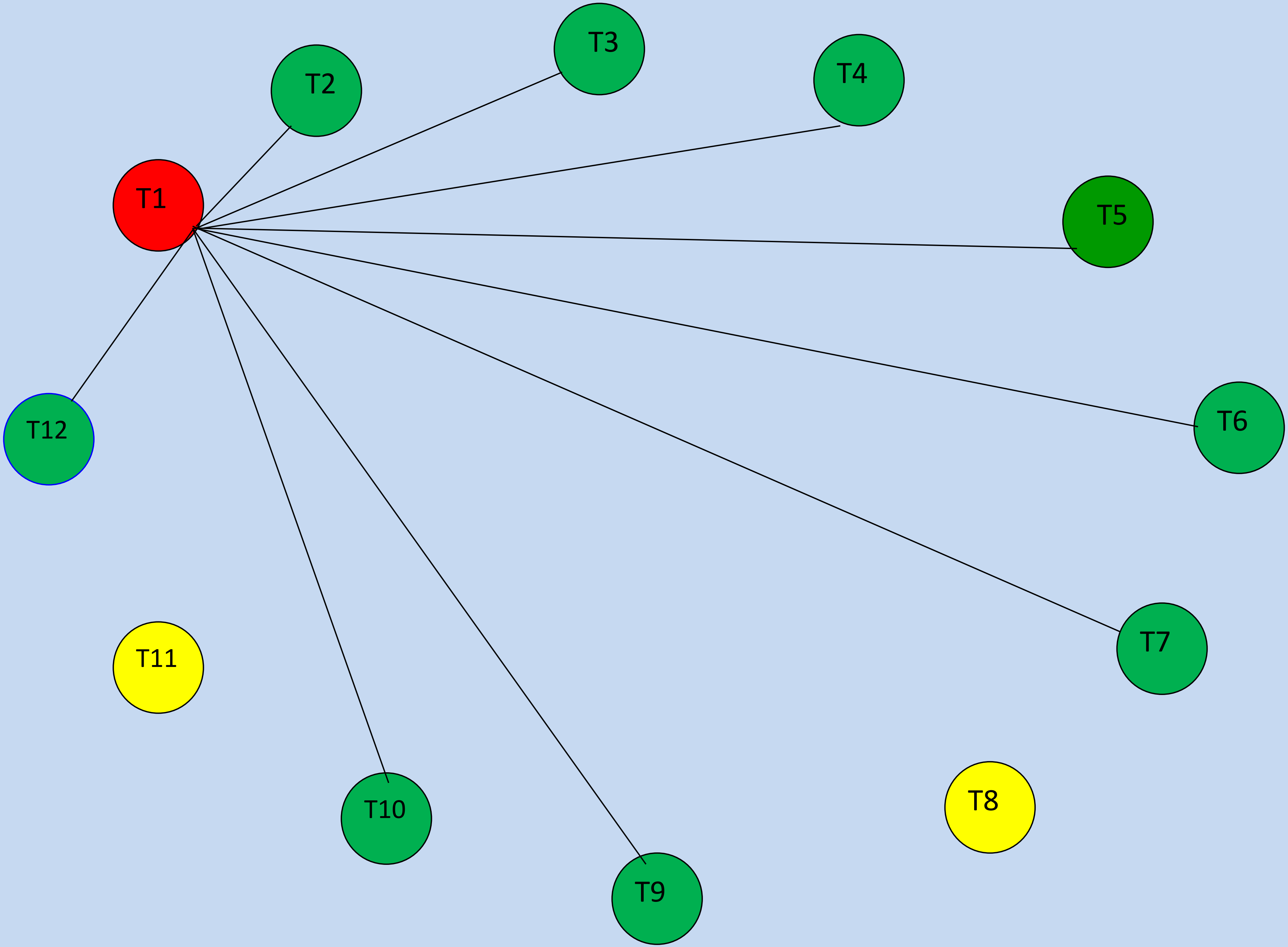
Le modèle proposé est un **graphe non orienté** G=(S,A), défini comme suit:

-l'ensemble S des **nœuds** représente l'ensemble des **tâches** de montage,

-les **arêtes** de l'ensemble A formalisent la **relation d'incompatibilité** entre tâches

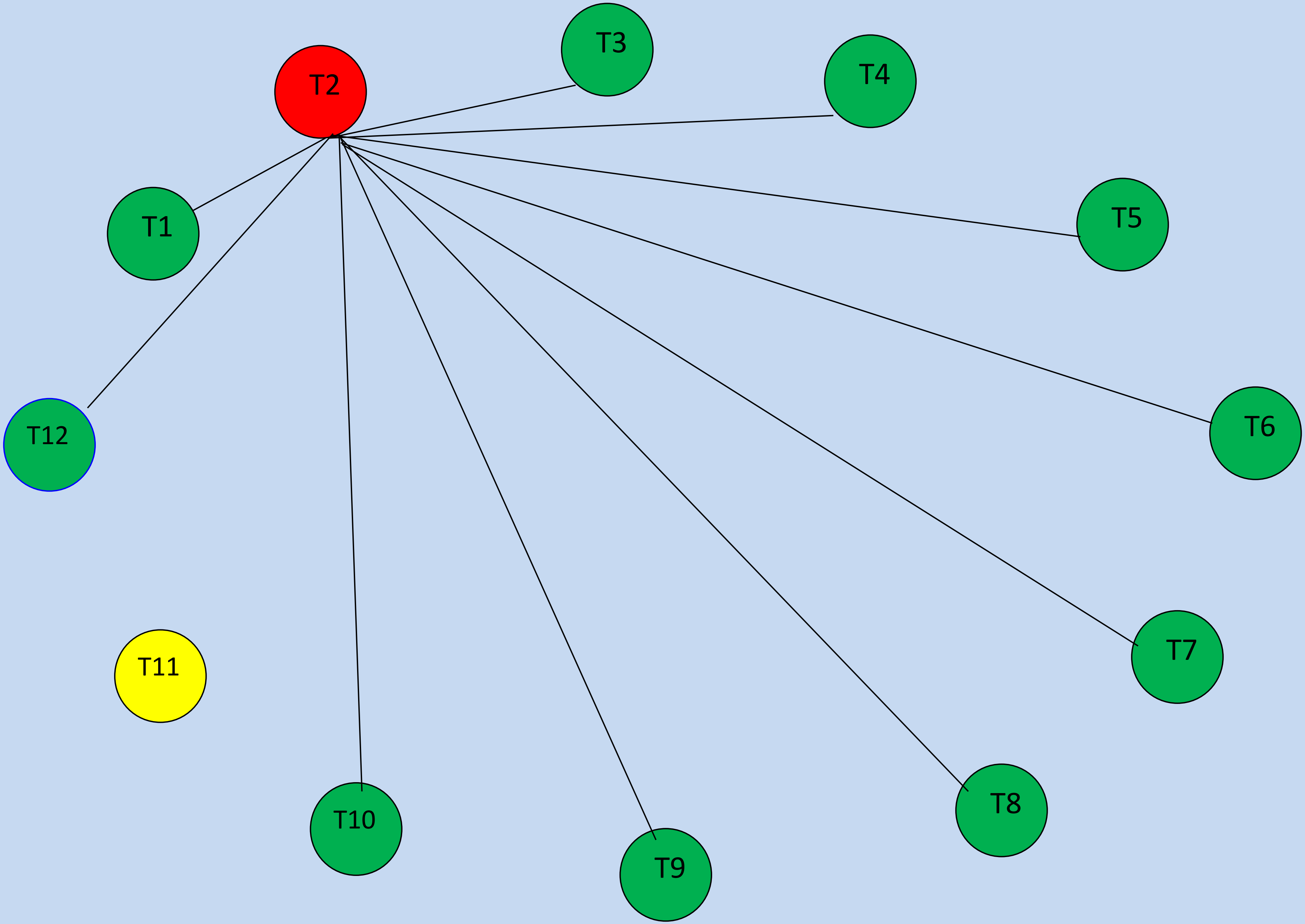

T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

La tâche	**T1**
…est incompatible avec les tâches	**T2** **T3** **T4** **T5** **T6** **T7** **T9** **T10** **T12**

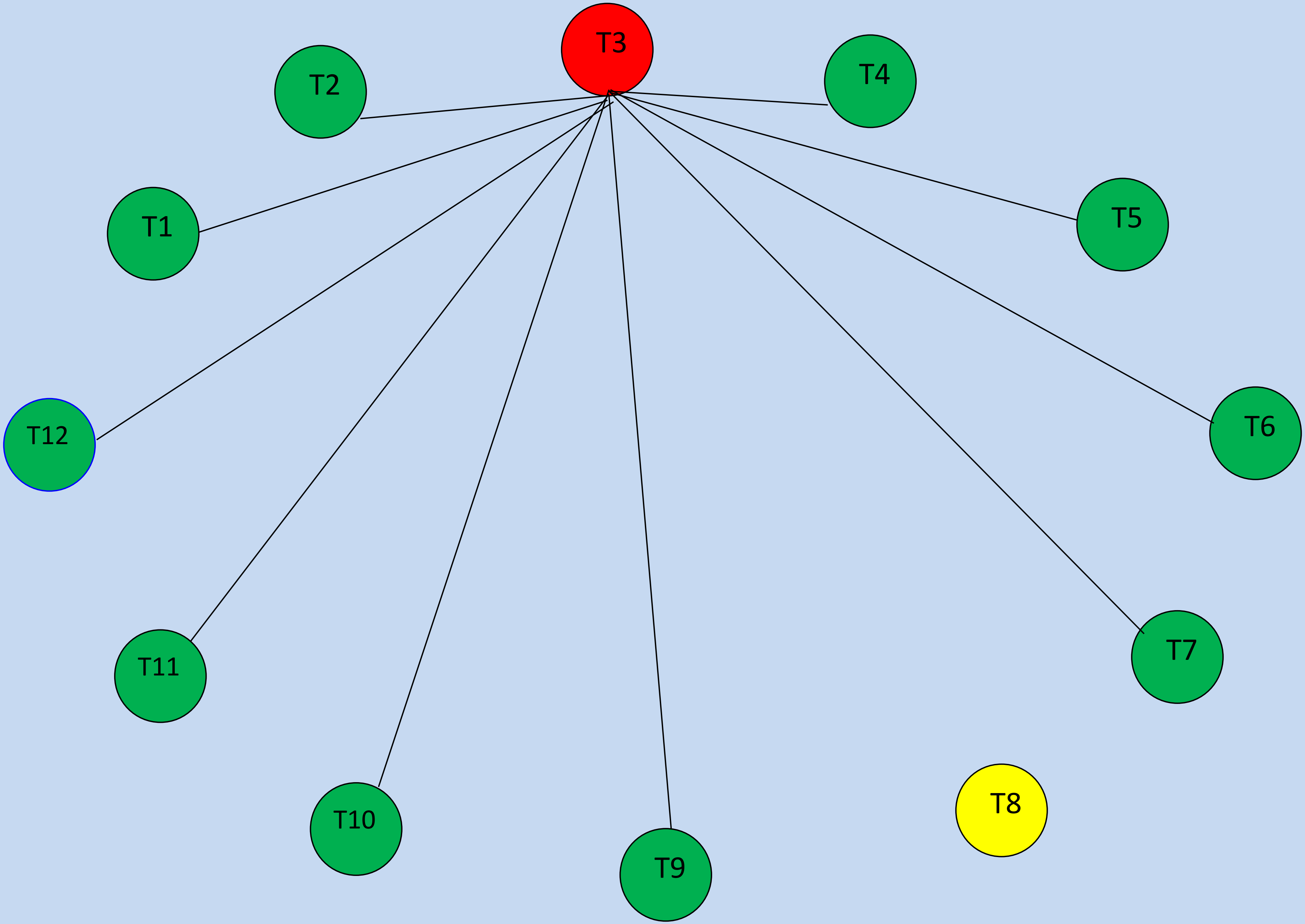


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

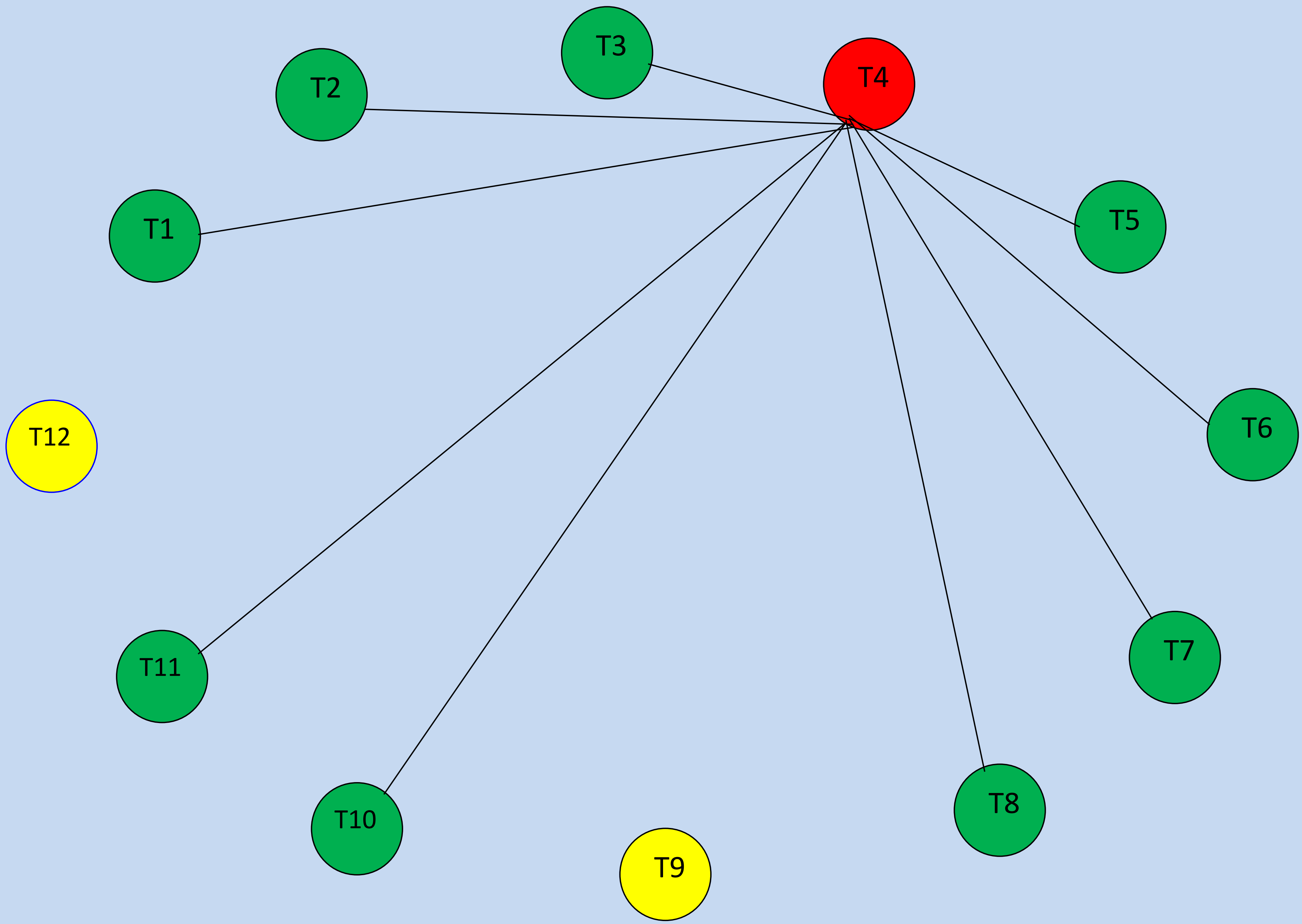
La tâche	**T2**
…est incompatible avec les tâches	**T1** **T3** **T4** **T5** **T6** **T7** **T8** **T9** **T10** **T12**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

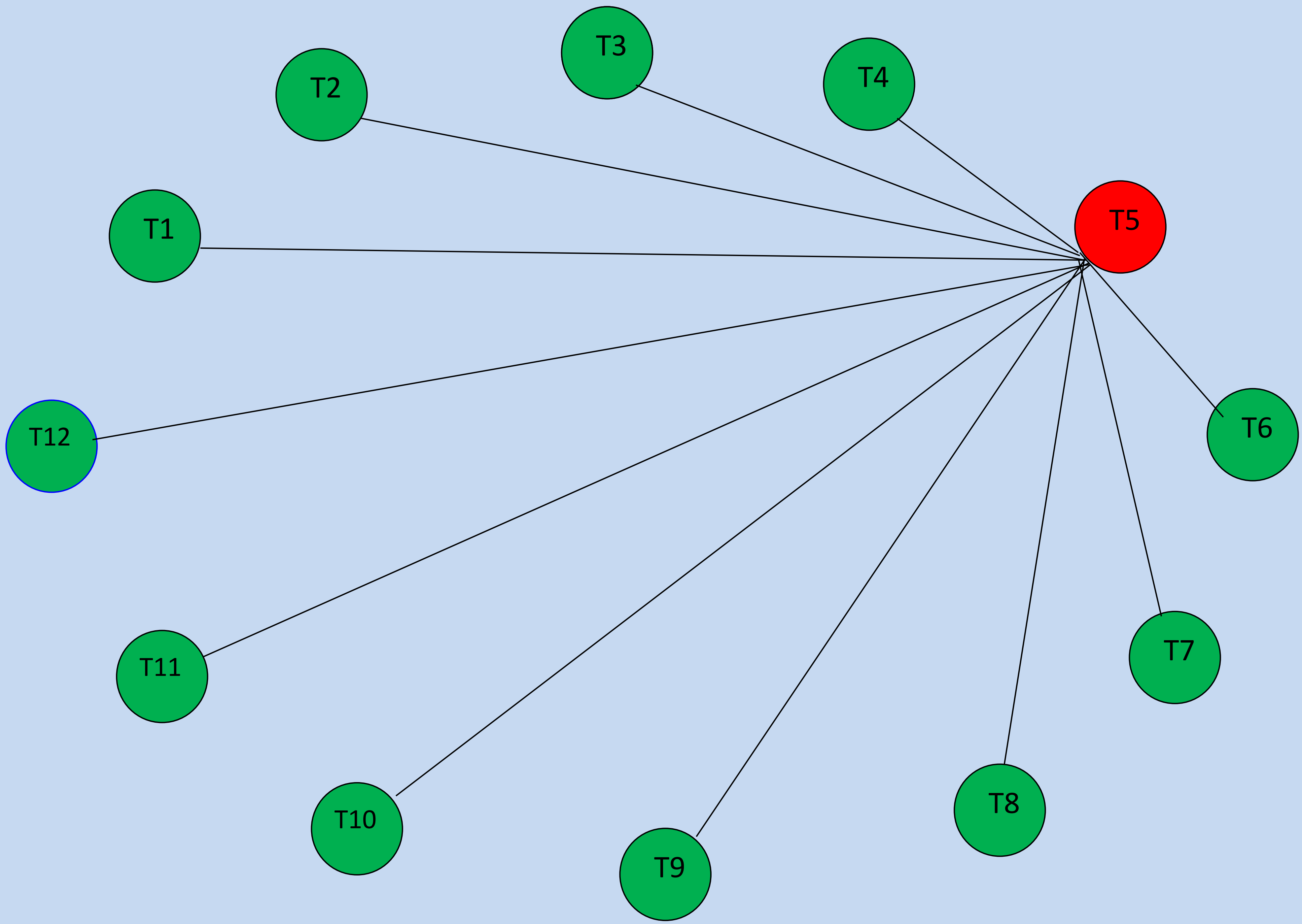
La tâche	**T3**
…est incompatible avec les tâches	**T1** **T2** **T4** **T5** **T6** **T7** **T9** **T10** **T11** **T12**


T3
T2
T4
T1
T5
T12
T6
T11
T7
T10
T9
T8

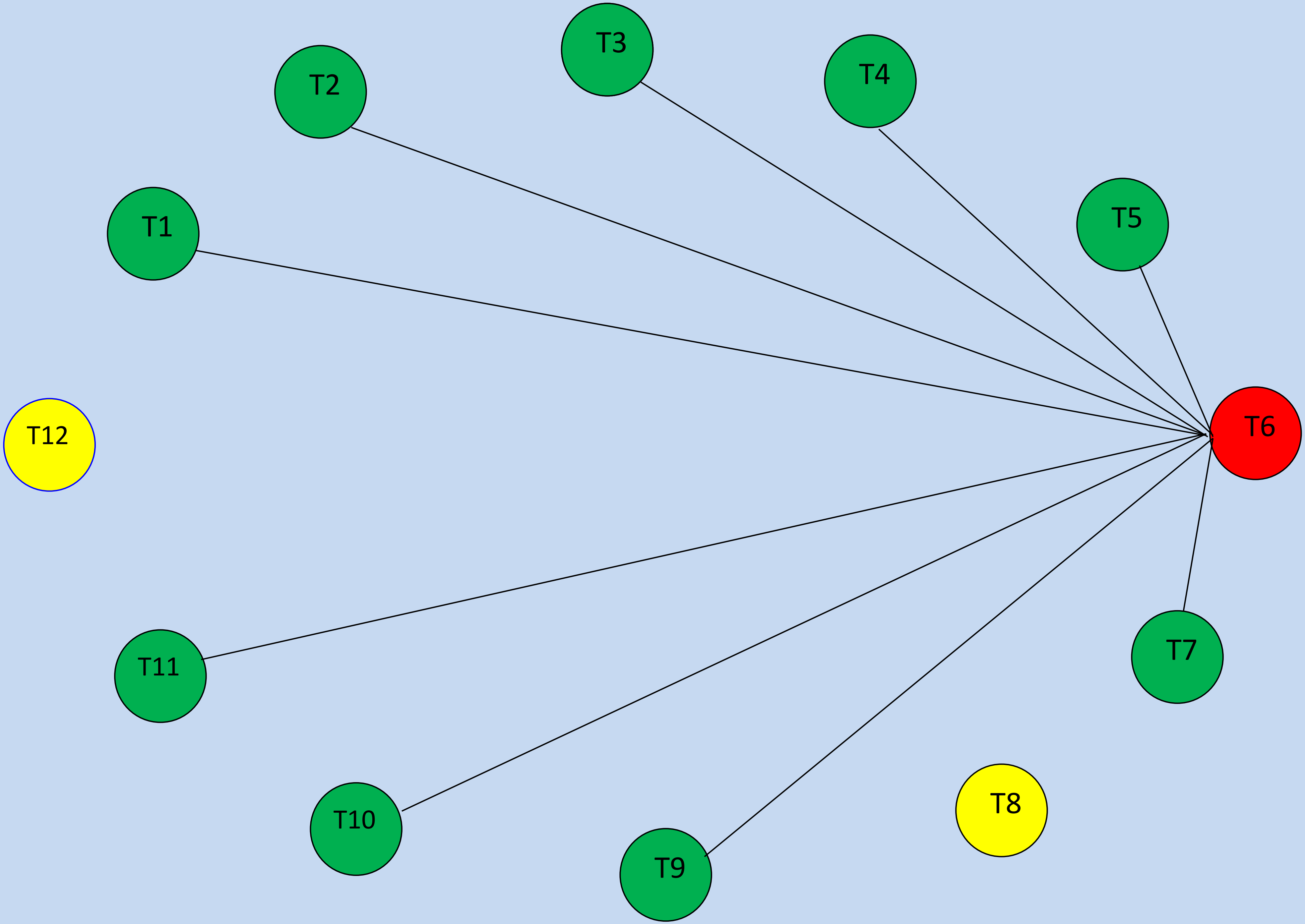
La tâche	**T4**
…est incompatible avec les tâches	**T1** **T2** **T3** **T5** **T6** **T7** **T8** **T10** **T11**


T3
T4
T2
T1
T5
T12
T6
T11
T7
T10
T8
T9

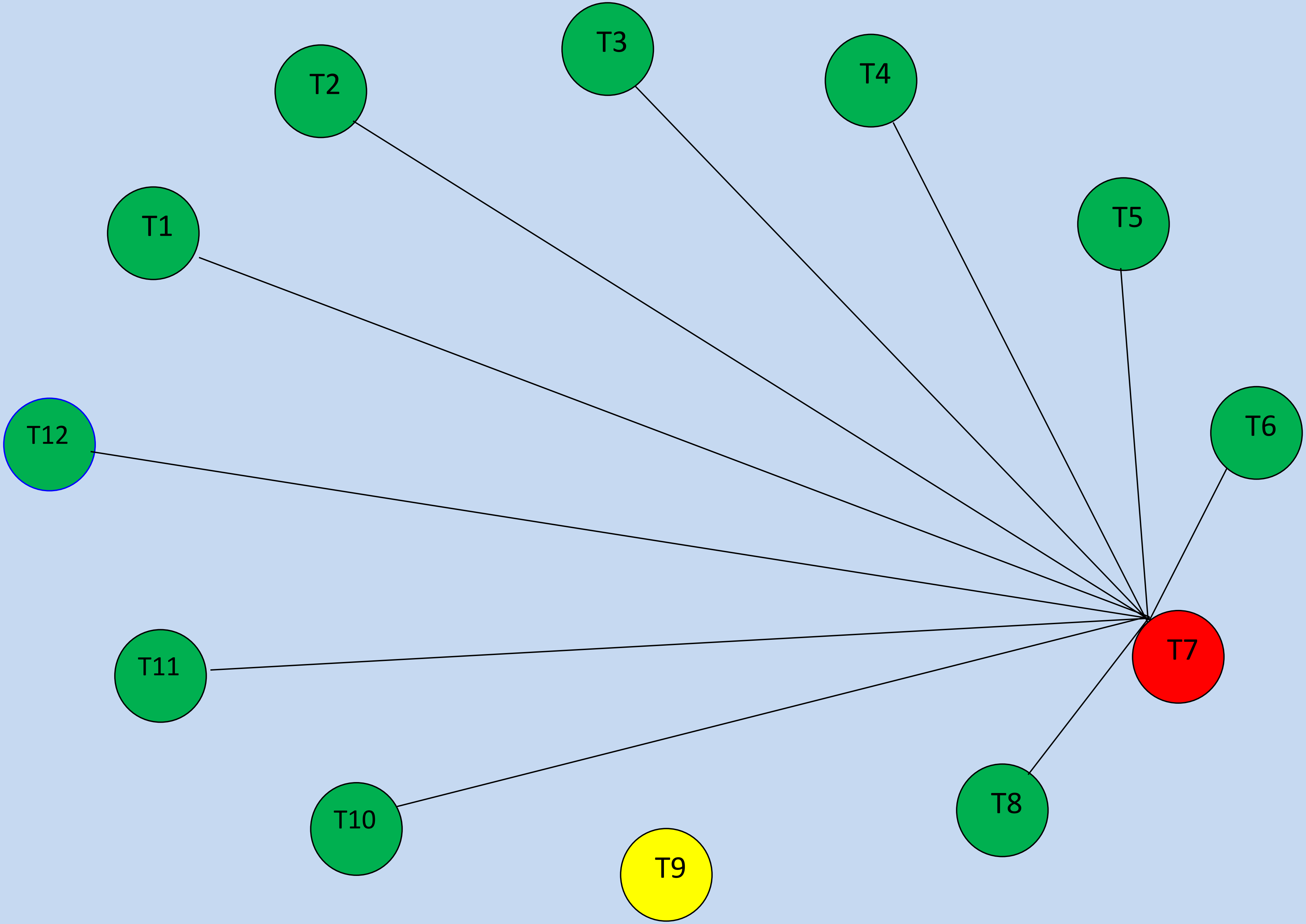
La tâche	**T5**
…est incompatible avec les tâches	**T1** **T2** **T3** **T4** **T6** **T7** **T8** **T9** **T10** **T11** **T12**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

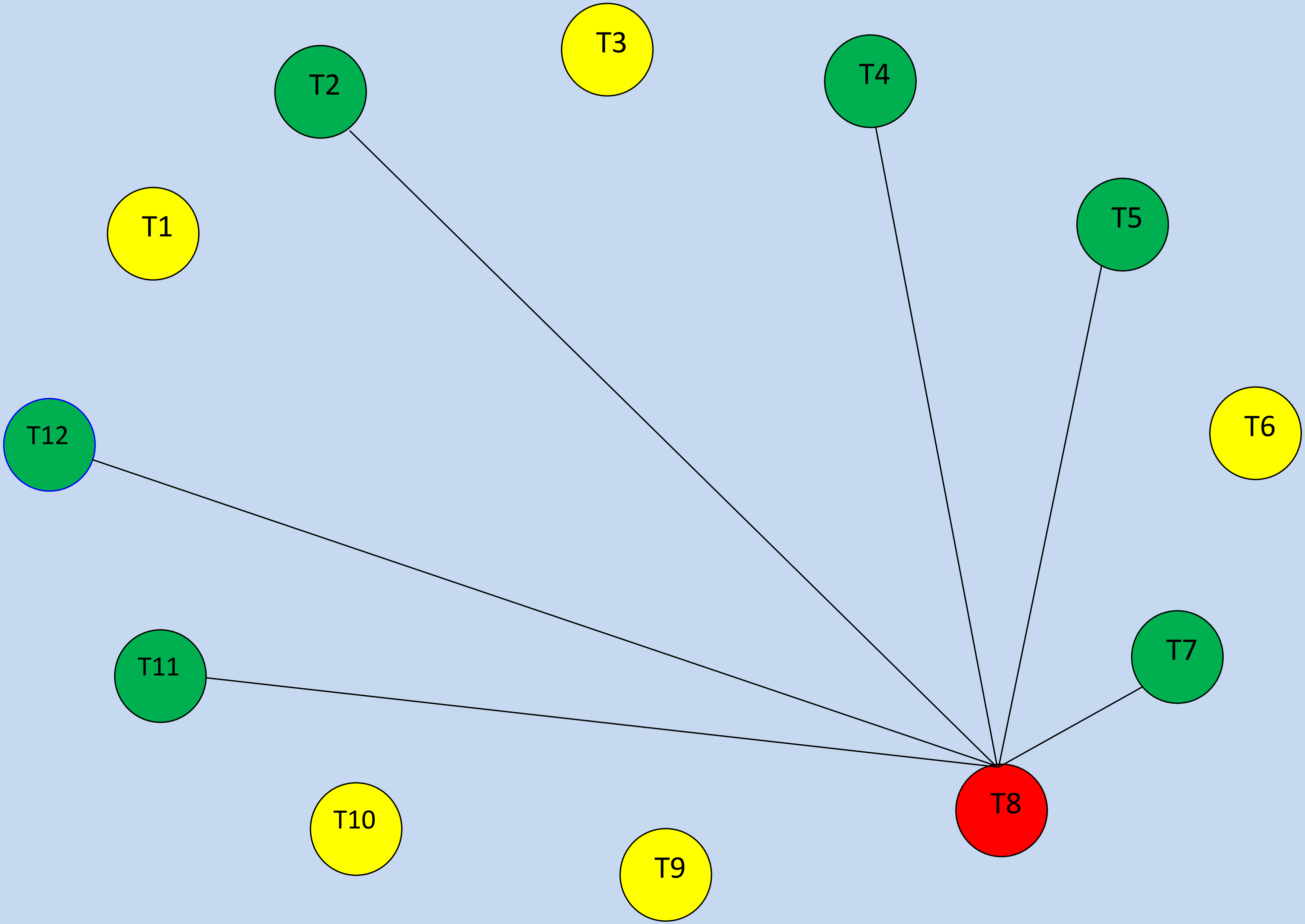
La tâche	**T6**
…est incompatible avec les tâches	**T1** **T2** **T3** **T4** **T5** **T7** **T9** **T10** **T11**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

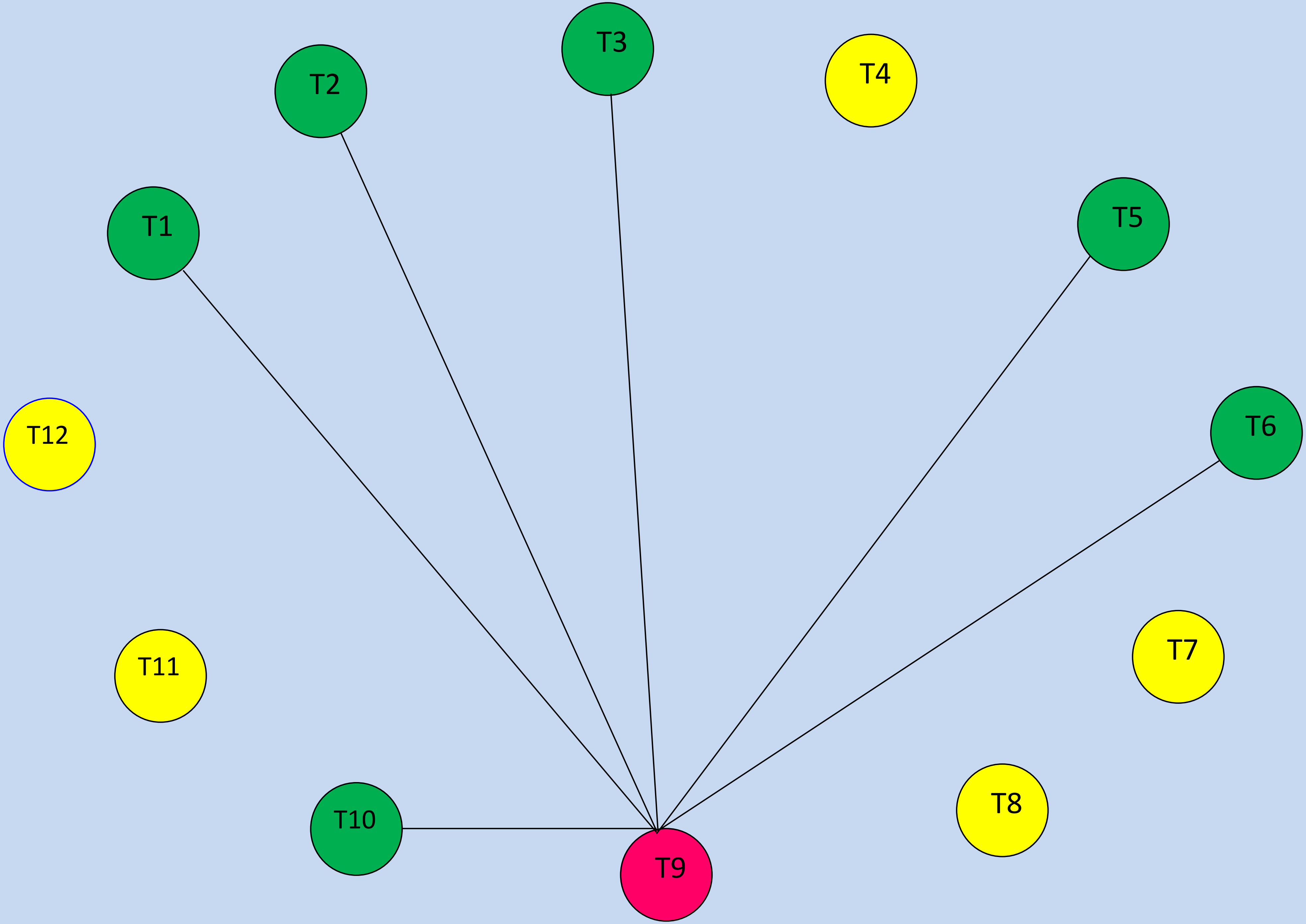
La tâche	**T7**
…est incompatible avec les tâches	**T1** **T2** **T3** **T4** **T5** **T6** **T8** **T10** **T11** **T12**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

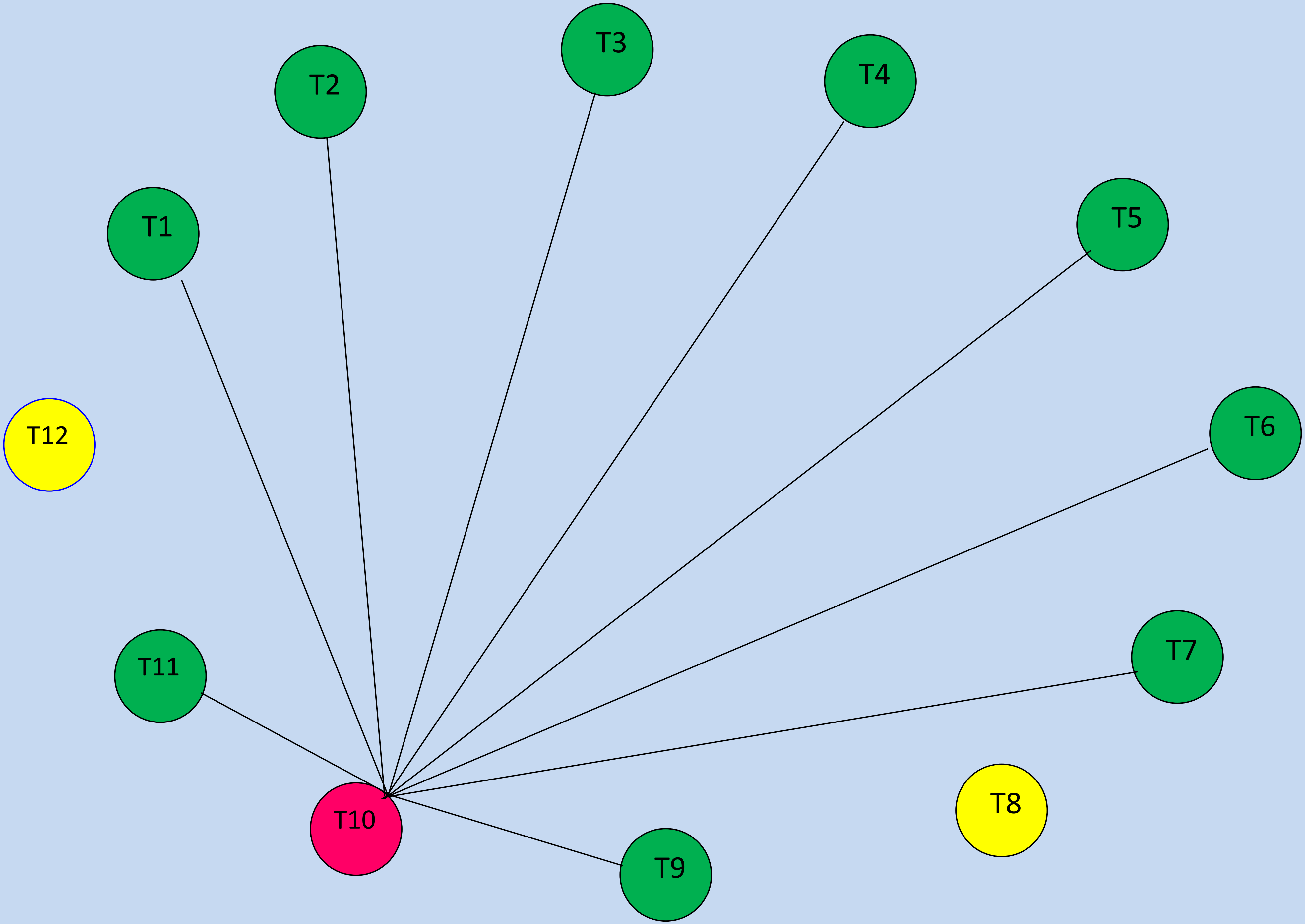
La tâche	**T8**
…est incompatible avec les tâches	**T2** **T4** **T5** **T7** **T11** **T12**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

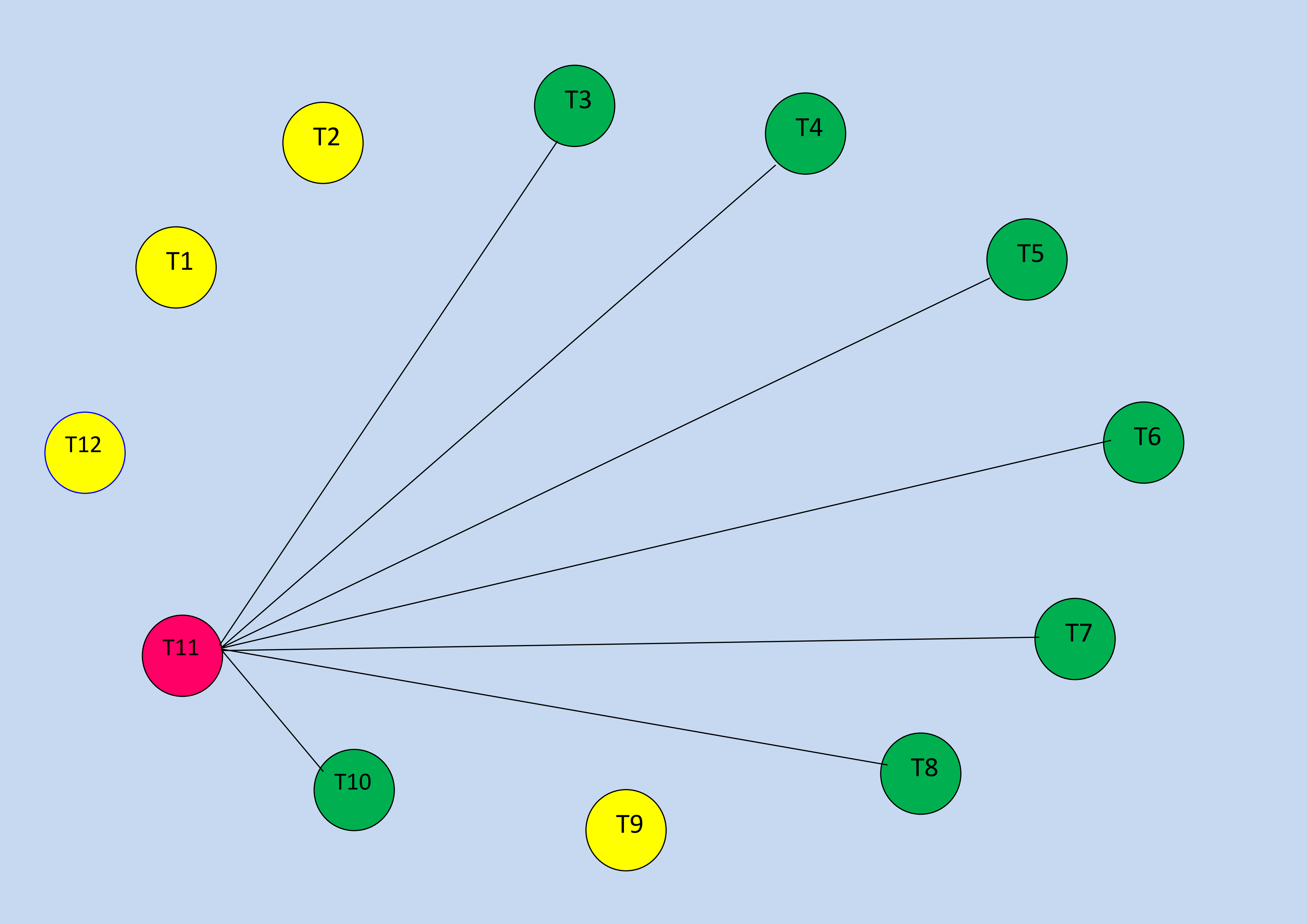
La tâche	**T9**
…est incompatible avec les tâches	**T1** **T2** **T3** **T5** **T6** **T10**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

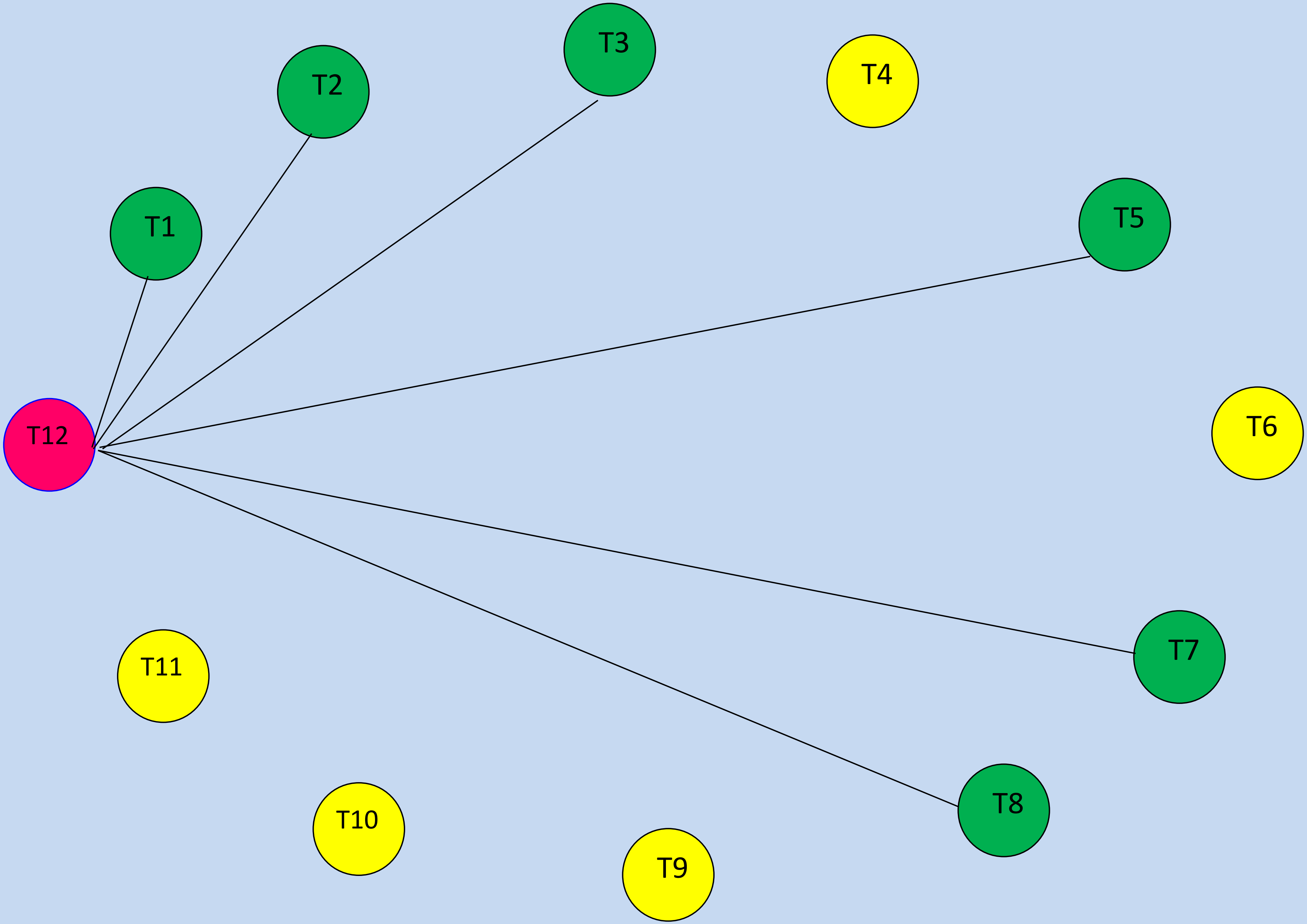
La tâche	**T10**
…est incompatible avec les tâches	**T1** **T2** **T3** **T4** **T5** **T6** **T7** **T9** **T11**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

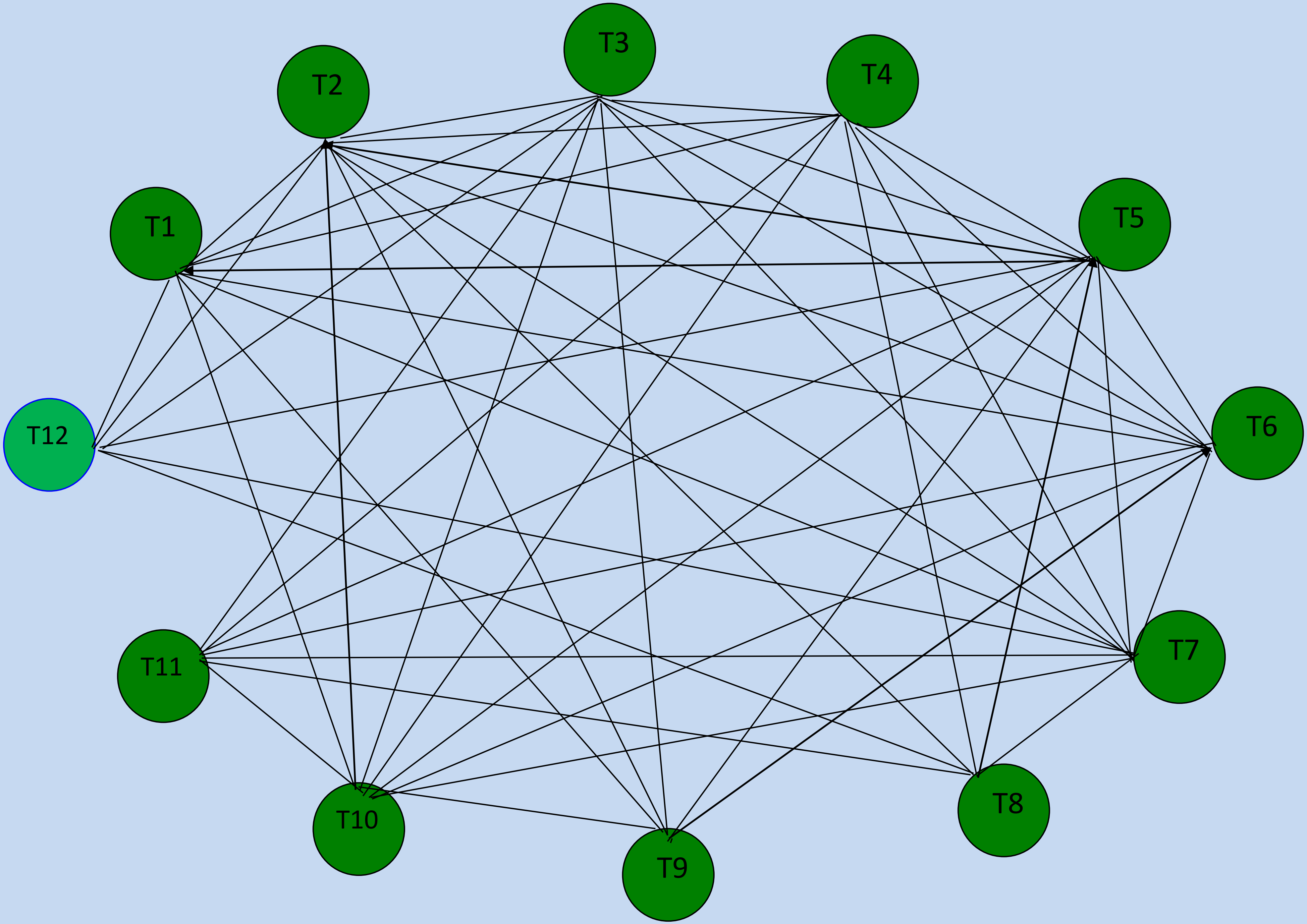
La tâche	**T11**
…est incompatible avec les tâches	**T3** **T4** **T5** **T6** **T7** **T8** **T10**


T3
T4
T2
T5
T1
T6
T12
T7
T11
T8
T10
T9

La tâche	**T12**
…est incompatible avec les tâches	**T1** **T2** **T3** **T5** **T7** **T8**


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

Tâche	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
incompatible avec les tâches	T2 T3 T4 T5 T6 T7 T9 T10 T12	T1 T3 T4 T5 T6 T7 T8 T9 T10 T12	T1 T2 T4 T5 T6 T7 T9 T10 T11 T12	T1 T2 T3 T5 T6 T7 T8 T10 T11	T1 T2 T3 T4 T6 T7 T8 T9 T10 T11 T12	T1 T2 T3 T4 T5 T7 T9 T10 T11	T1 T2 T3 T4 T5 T6 T8 T10 T11 T12	T2 T4 T5 T7 T11 T12	T1 T2 T3 T5 T6 T10	T1 T2 T3 T4 T5 T6 T7 T9 T11	T3 T4 T5 T6 T7 T8 T10	T1 T2 T3 T5 T7 T8


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

2- Montrer que le problème d’optimisation du cycle peut être formulé en termes d’un problème classique de coloration de graphe.

Le problème consiste à ordonnancer les tâches de sorte à maximiser la «parallélisation» des tâches.

Cela revient à mettre en évidence dans le cycle de montage le **nombre minimum** de tâches incompatibles.

Si on applique une coloration au modèle de graphe d'incompatibilité, les nœuds représentant des tâches incompatibles sont adjacents donc porteur des **couleurs différentes**.

Le problème peut être formulé en termes d'un problème classique de recherche du **nombre minimum** (appelé aussi nombre chromatique) pour colorier un graphe.

Sur le modèle de graphe, les nœuds portant une <u>même couleur</u> représentent des tâches <u>non incompatibles</u> : tâches qui peuvent être exécutées en parallèle (car elles ne préemptent pas les mêmes robots).

L’optimisation du cycle de montage se ramène donc au calcul du **nombre chromatique** du modèle de graphe d’incompatibilité.

3. Proposer une solution à ce problème: exposer la procédure utilisée et la trace d’exécution correspondante

Le problème de calcul du nombre chromatique est un problème **NP-complet**.

L’algorithme de Welsh Powell est un algorithme **polynomial** qui fournit une solution «**approchée**».

Il permet d'obtenir une coloration du graphe qui utilise un nombre k «**pas trop grand»** de couleurs.

Cependant il **n'assure pas** que le résultat retourné soit **minimum**, c’est à dire que :

$$k = \gamma(G)$$

avec $\gamma(G)$ désignant le nombre chromatique de G.

La mise en œuvre de l'algorithme implique trois étapes :

<u>Etape1</u>:

1- Classer les nœuds du graphe dans l'ordre **décroissant** de leur degré.

2- Attribuer à chacun des nœuds un ordre dans la liste triée.

Étape 2

En parcourant la liste des nœuds **dans l'ordre** de tri:

1- attribuer une couleur **non encore utilisée** au premier nœud **non encore coloré**

2- attribuer cette **même couleur** à chaque nœud **non encore coloré** et **non adjacent** à un nœud de **cette couleur**.

Étape 3

1-S'il reste des nœuds non colorés revenir à l'étape 2.

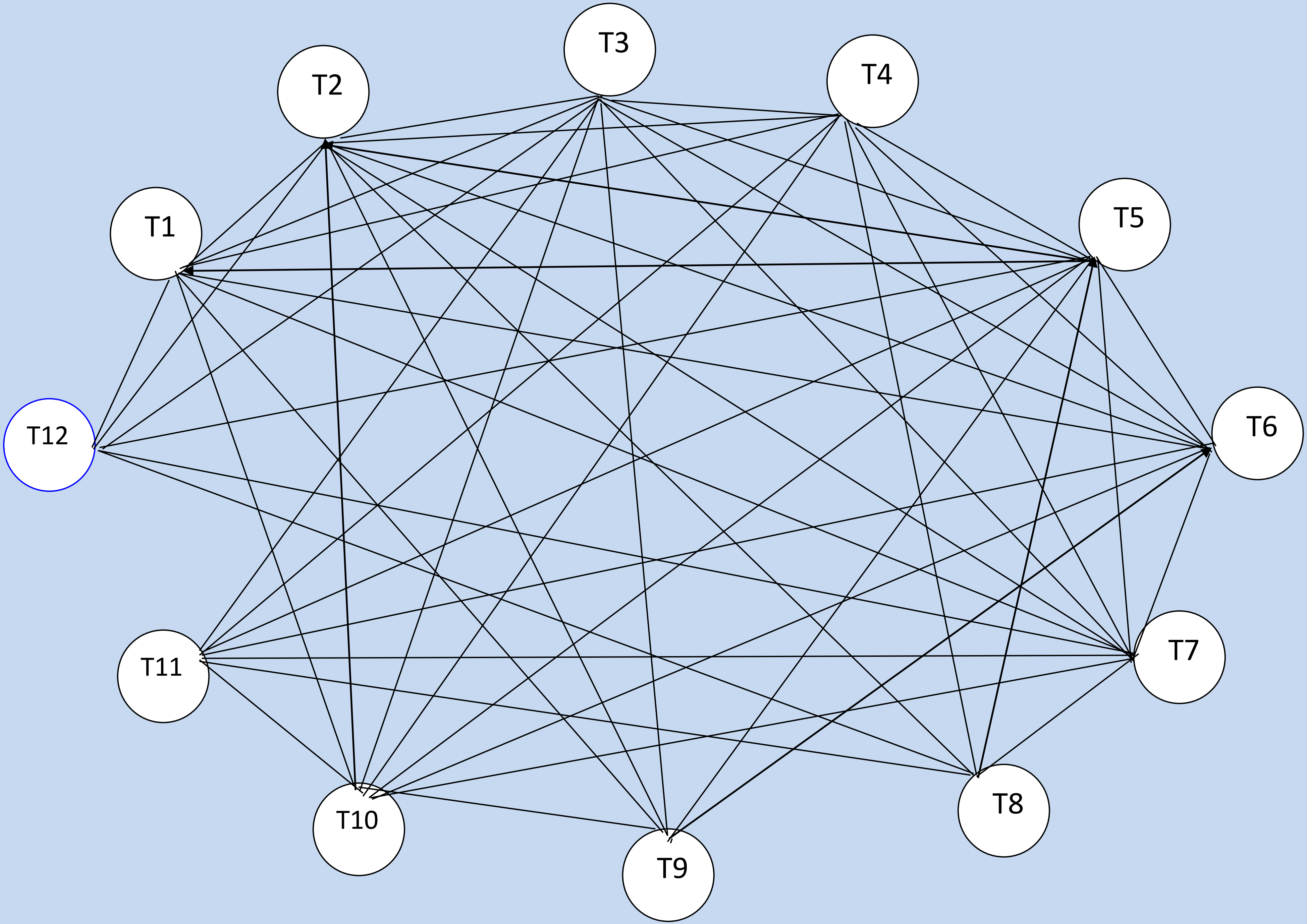
2- Sinon, la coloration des nœuds est terminée.

Mise en œuvre de l’algorithme de W-P :

Au départ :

-le graphe G, modèle d’incompatibilité des t

-aucun nœud n’est coloré


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

Etape 1 : commencer par trier la liste des 12 nœuds dans l'ordre **décroissant** de leur degré.

Tâche	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
degré	9	10	10	9	11	9	10	6	6	9	7	6
incompatible avec les tâches	T2 T3 T4 T5 T6 T7 T9 T10 T12	T1 T3 T4 T5 T6 T7 T8 T9 T10 T12	T1 T2 T4 T5 T6 T7 T9 T10 T11 T12	T1 T2 T3 T5 T6 T7 T8 T10 T11	T1 T2 T3 T4 T6 T7 T8 T9 T10 T11 T12	T1 T2 T3 T4 T5 T7 T9 T10 T11	T1 T2 T3 T4 T5 T6 T8 T10 T11 T12	T2 T4 T5 T7 T11 T12	T1 T2 T3 T5 T6 T10	T1 T2 T3 T4 T5 T6 T7 T9 T11	T3 T4 T5 T6 T7 T8 T10	T1 T2 T3 T5 T7 T8

1- $d°(T5) = 11$

2- $d°(T2) = 10$

3- $d°(T3) = 10$

4- $d°(T7) = 10$

5- $d°(T1) = 9$

6- $d°(T4) = 9$

7- $d°(T6) = 9$

8- $d°(T10) = 9$

9- $d°(T11) = 7$

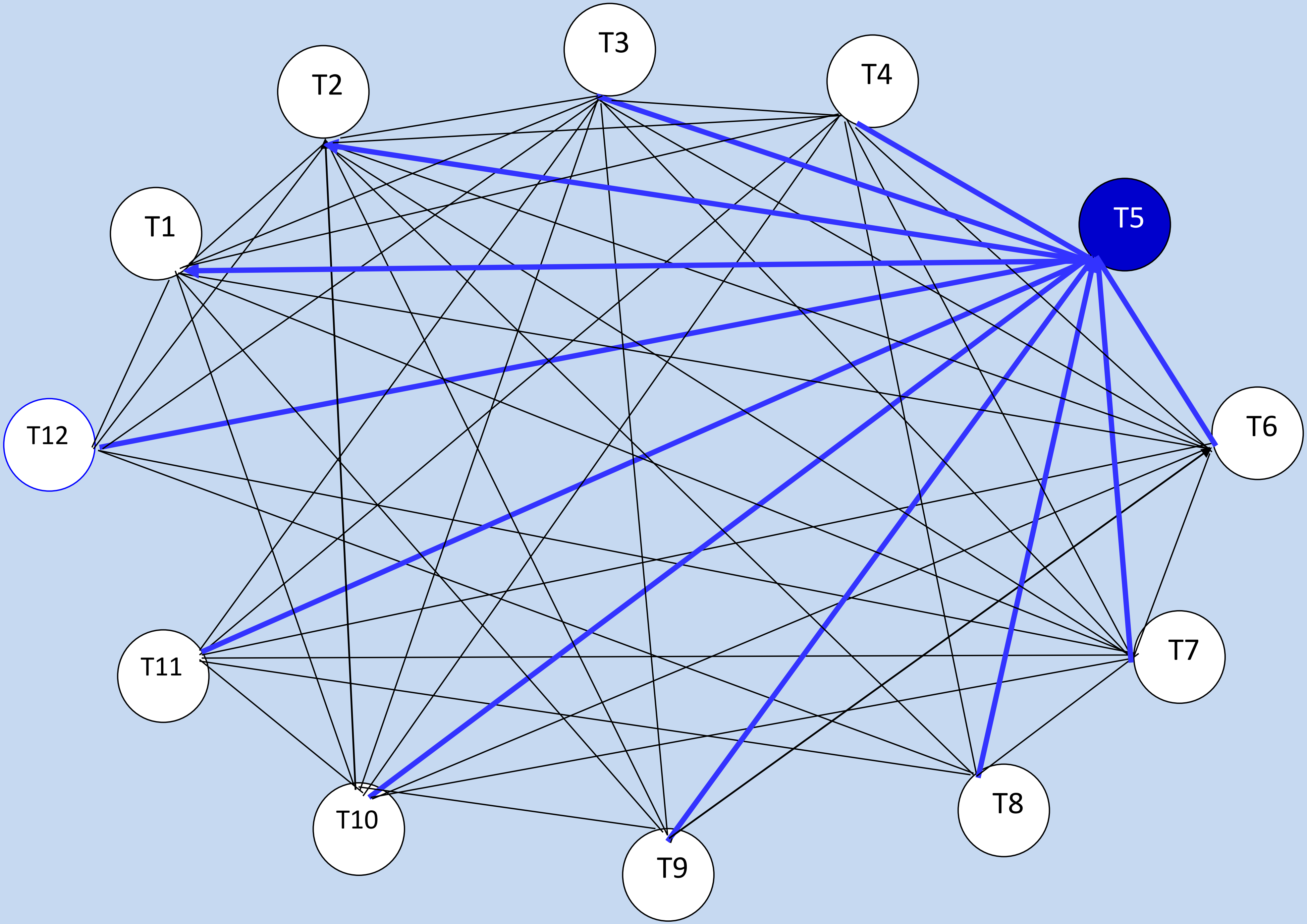
10- $d°(T8) = 6$

11- $d°(T9) = 6$

12- $d°(T12) = 6$

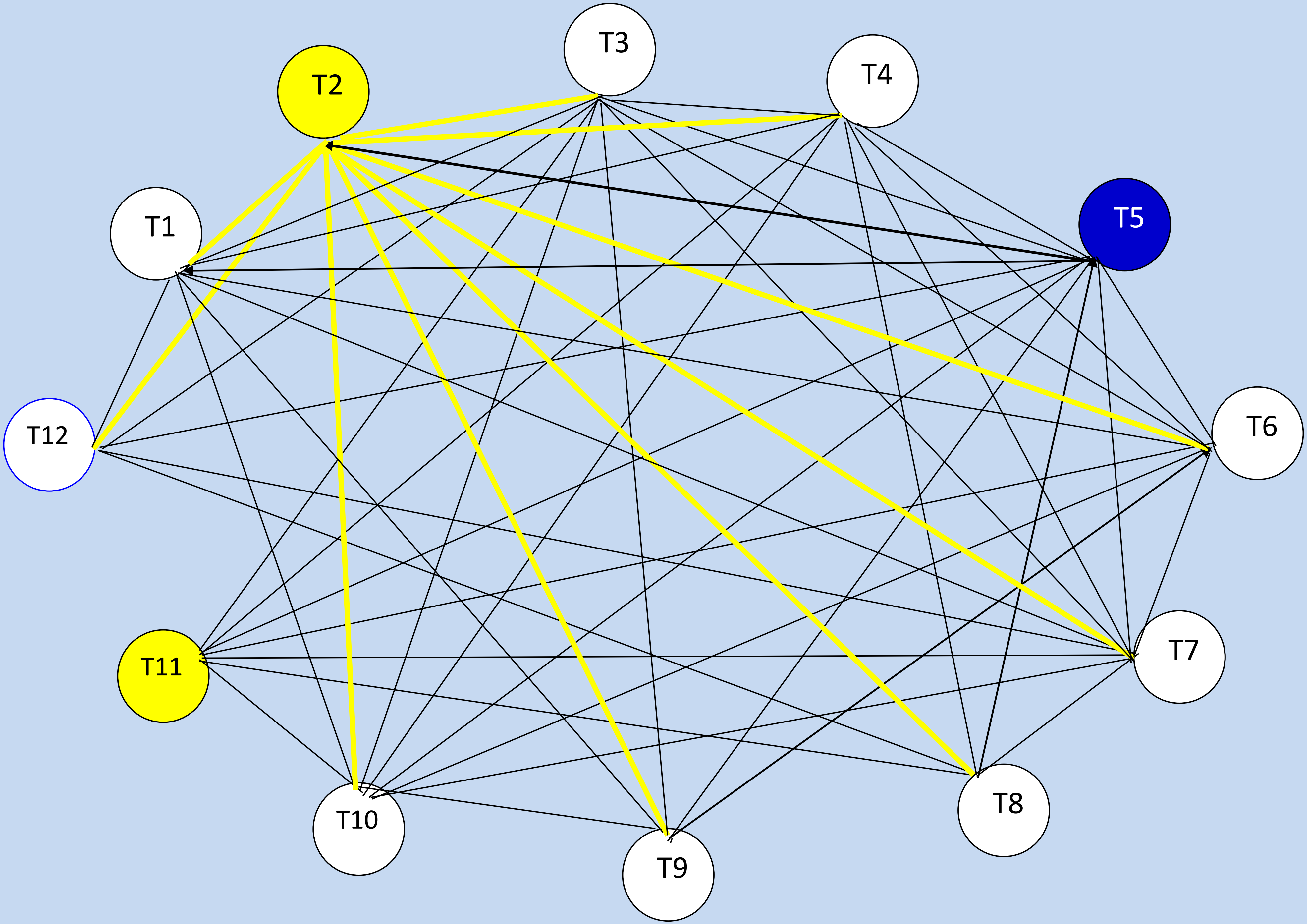
Etape 2 : colorier en bleu, par exemple, le nœud **T5** et tous les nœuds non adjacents

13- d°(T5) = 11
14- d°(T2) = 10
15- d°(T3) = 10
16- d°(T7) = 10
17- d°(T1) = 9
18- d°(T4) = 9
19- d°(T6) = 9
20- d°(T10) = 9
21- d°(T11) = 7
22-d°(T8) = 6
23-d°(T9) = 6
24-d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

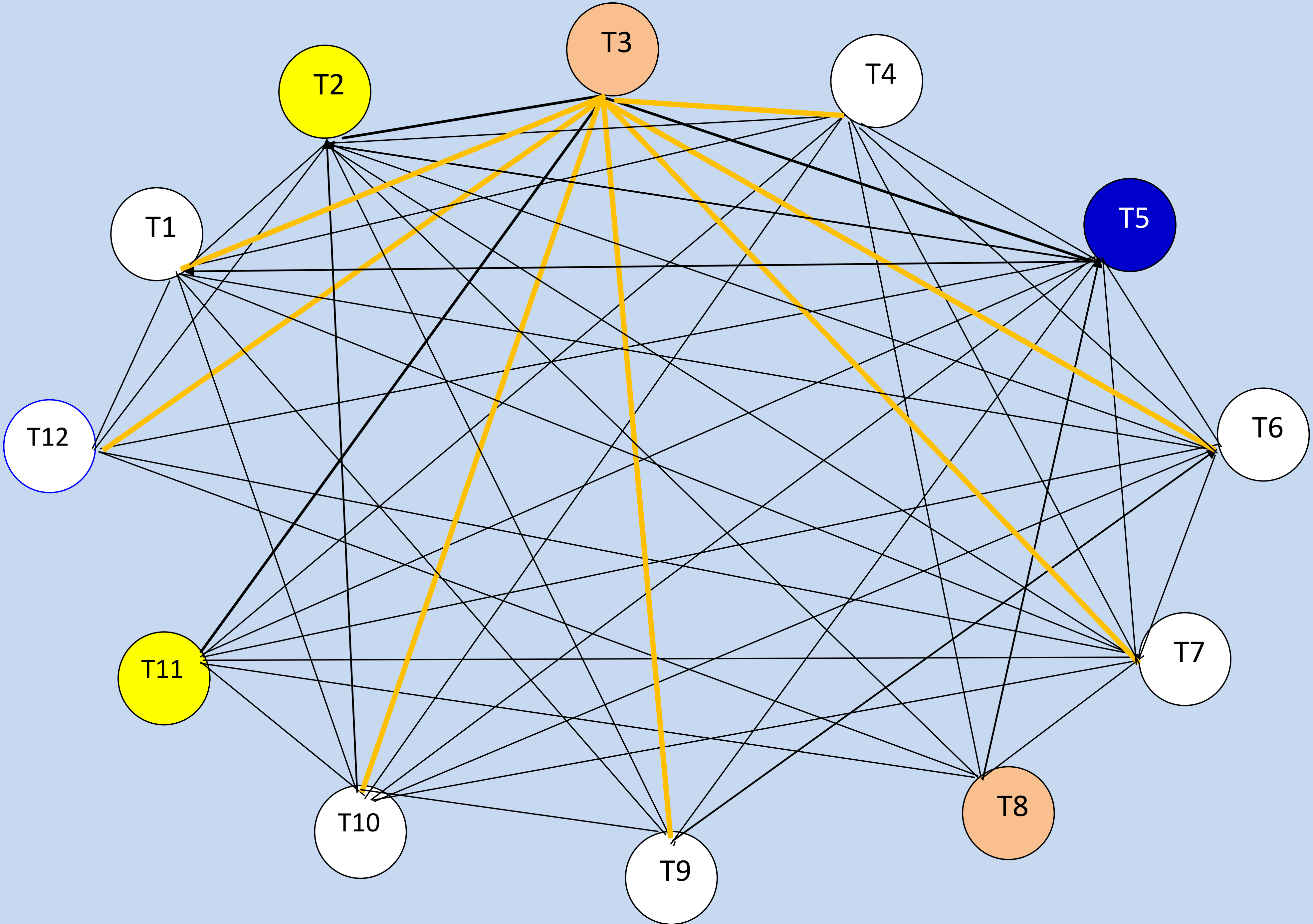
Etape 3 : revenir à l'étape 2 et colorier en jaune le nœud T2 et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10- d°(T8) = 6
11- d°(T9) = 6
12- d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

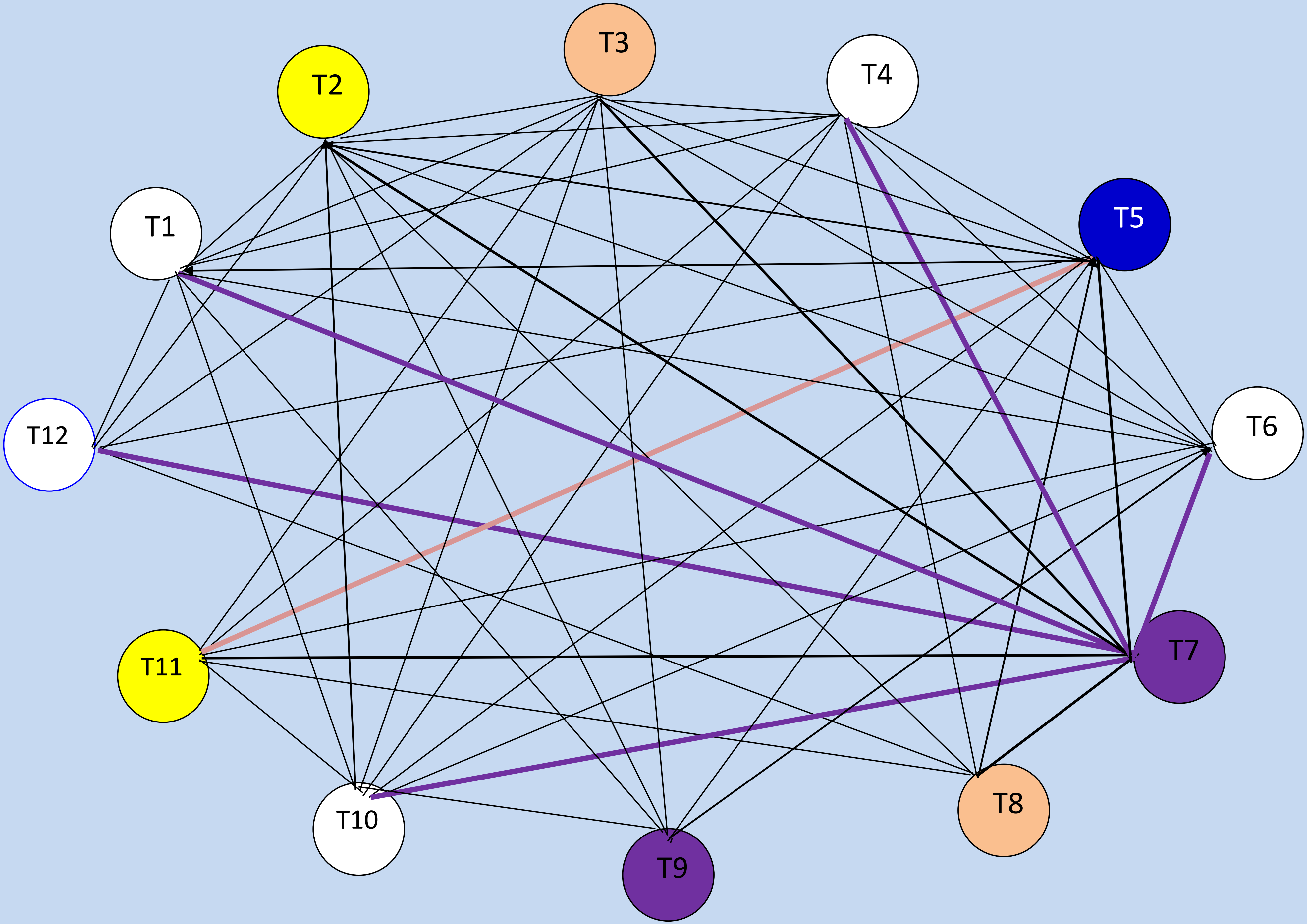
Etape 3 : revenir à l’étape 2 et colorier en orange le nœud T3 et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10-d°(T8) = 6
11-d°(T9) = 6
12-d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

Etape 3 : revenir à l'étape 2 et colorier en violet le nœud T7 et tous les nœuds non coloriés non adjacents

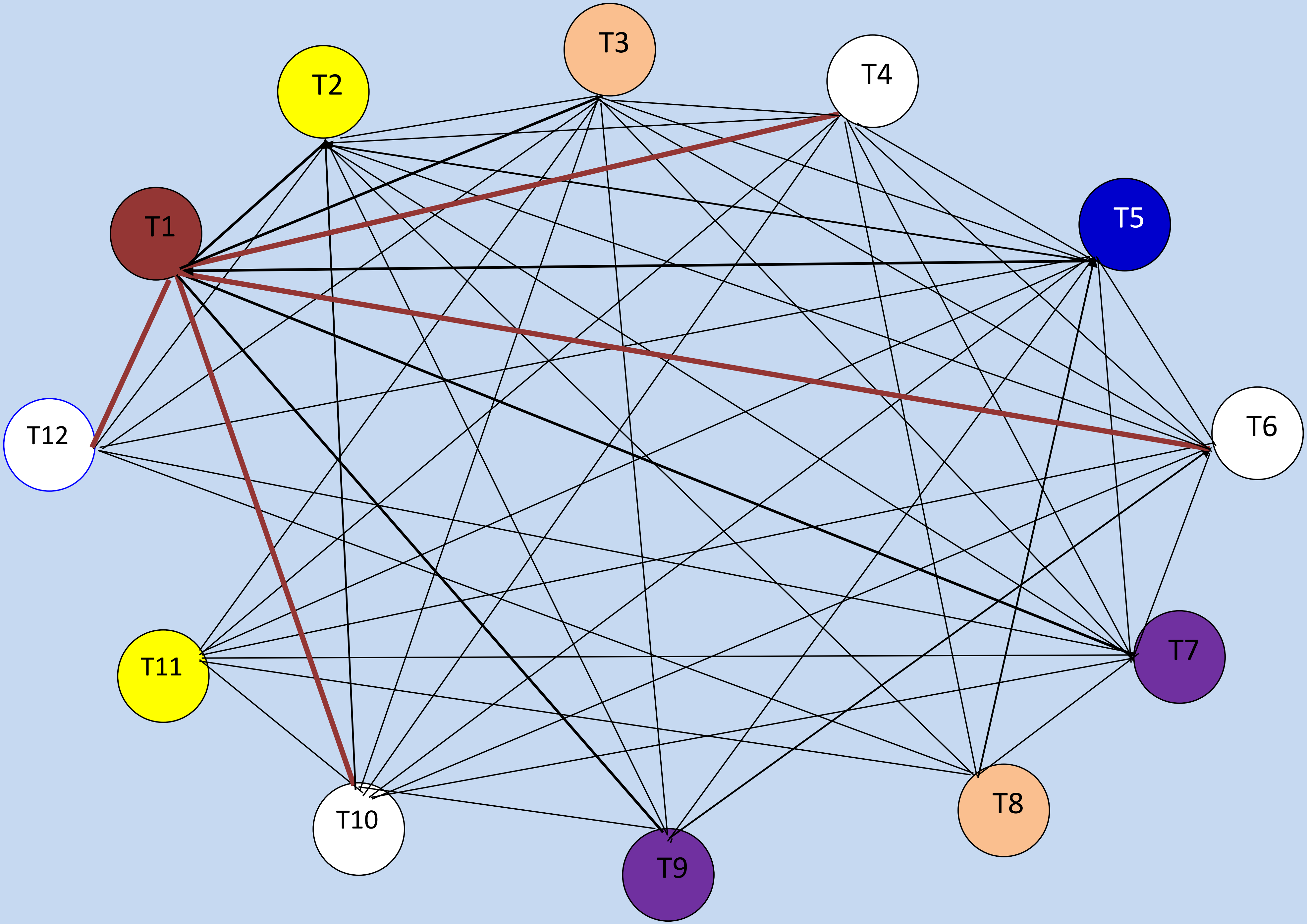
1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10-d°(T8) = 6
11-d°(T9) = 6
12-d°(T12) = 6



T3
T2
T4
T1
T5
T12
T6
T11
T7
T10
T9
T8

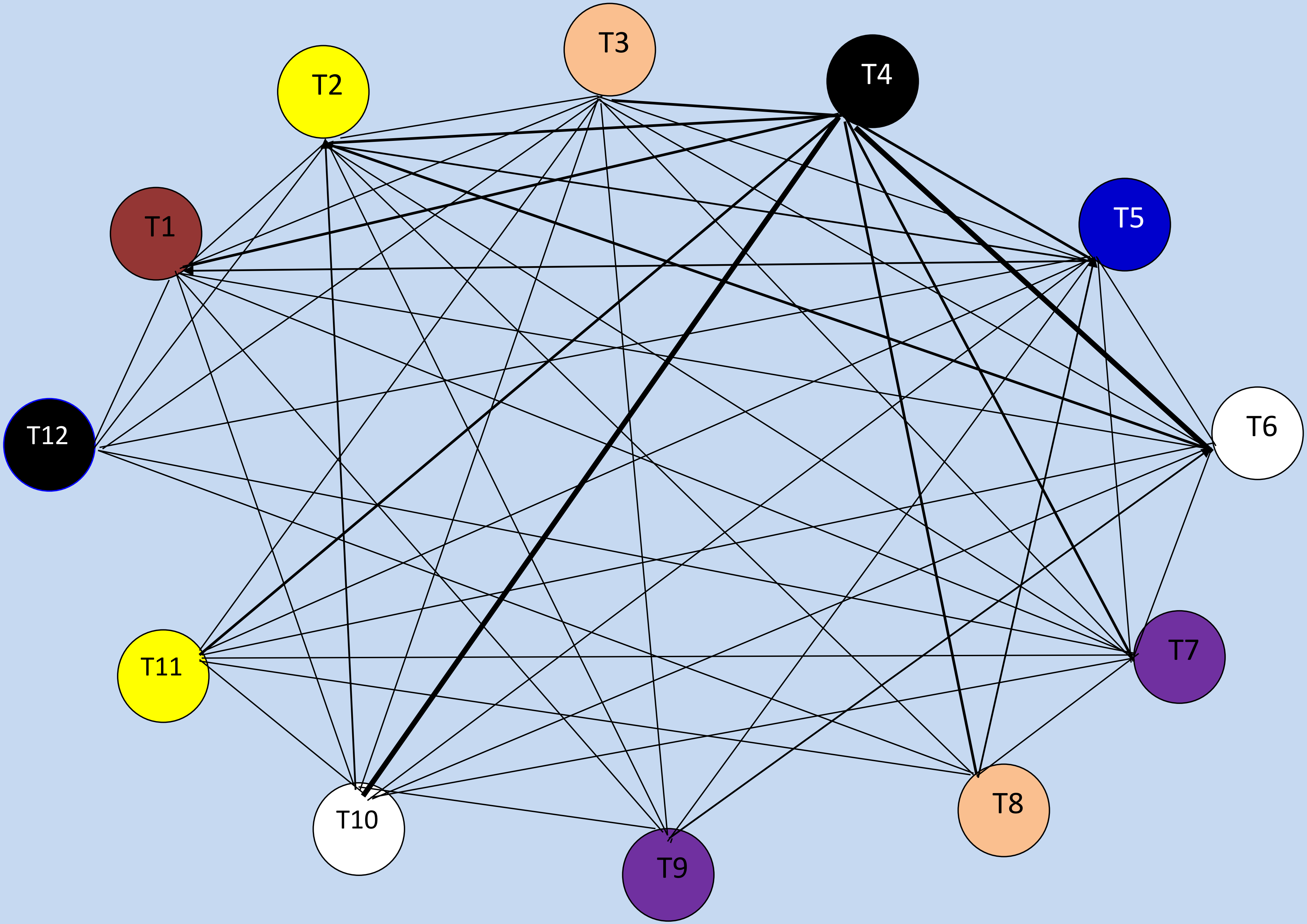
Etape 3 : revenir à l'étape 2 et colorier en prune le nœud T1 et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10-d°(T8) = 6
11-d°(T9) = 6
12-d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

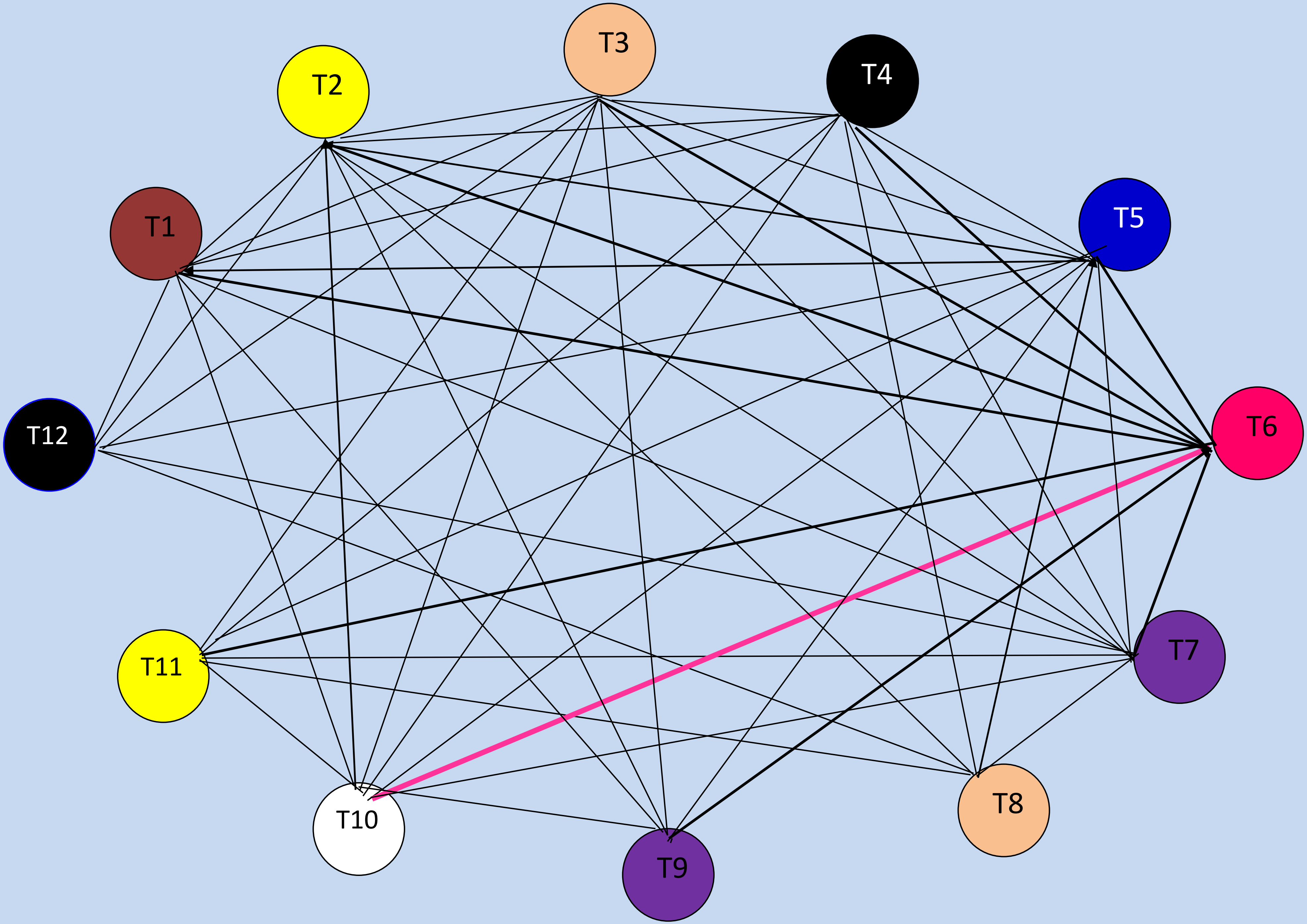
Etape 3 : revenir à l'étape 2 et colorier en noir le nœud **T4** et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10-d°(T8) = 6
11-d°(T9) = 6
12-d°(T12) = 6


T3
T4
T2
T5
T1
T6
T12
T7
T11
T8
T10
T9

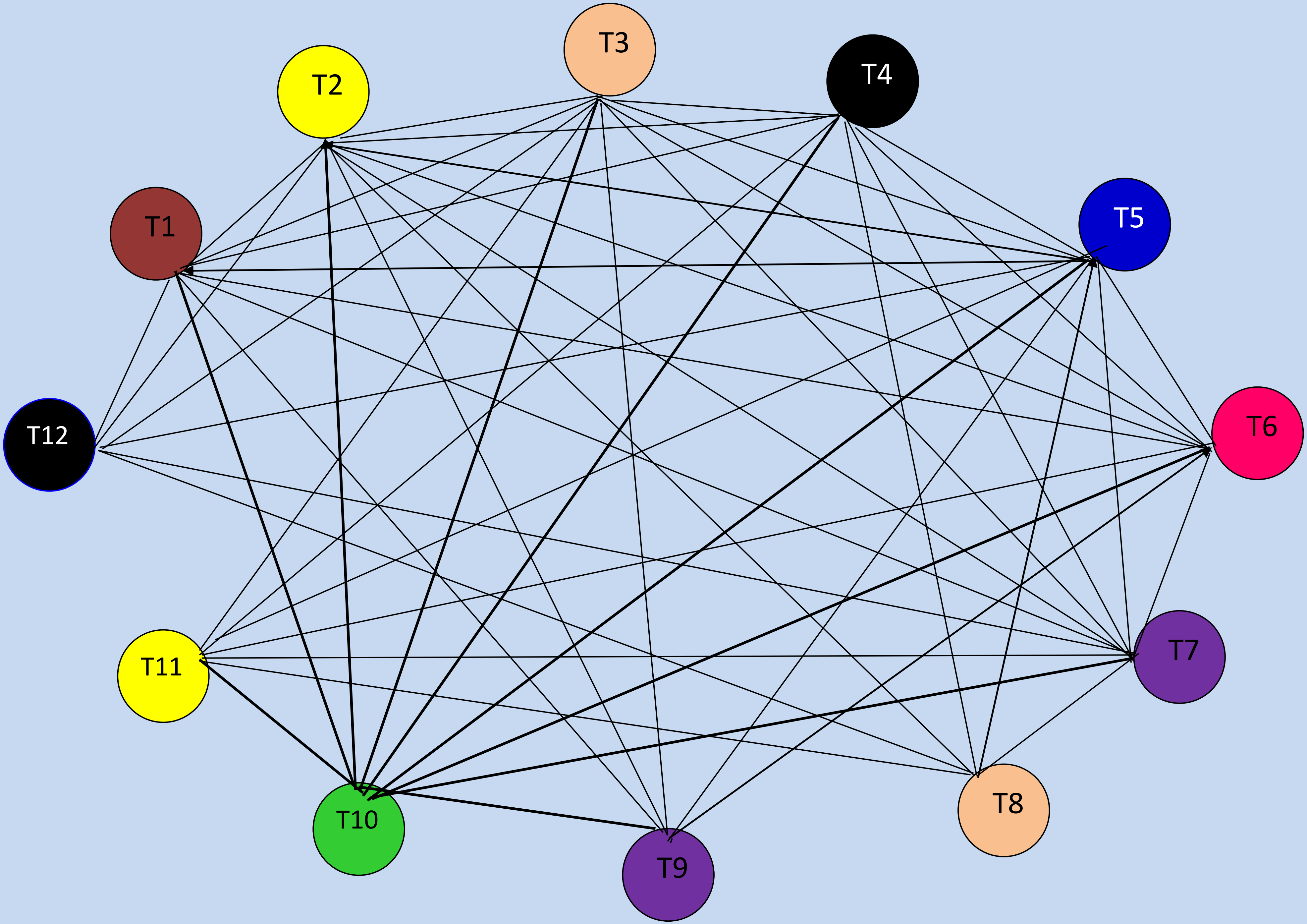
Etape 3 : revenir à l'étape 2 et colorier en cerise le nœud **T6** et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10- d°(T8) = 6
11- d°(T9) = 6
12- d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

Etape 3 : revenir à l'étape 2 et colorier en vert le nœud **T10** et tous les nœuds non coloriés non adjacents

1- d°(T5) = 11
2- d°(T2) = 10
3- d°(T3) = 10
4- d°(T7) = 10
5- d°(T1) = 9
6- d°(T4) = 9
7- d°(T6) = 9
8- d°(T10) = 9
9- d°(T11) = 7
10- d°(T8) = 6
11- d°(T9) = 6
12- d°(T12) = 6


T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
T11
T12

Etape 3 : tous les nœuds sont coloriés, l'algorithme s'arrête

k = 8 : résultat de l'algorithme de Welsh-Powell

$$\gamma(G) \approx 8$$

Vérification d'admissibilté:

$$\gamma(G) \leq r+1 = d°(5) +1 = 11+1= 12$$

$$\gamma(G) \leq n+1 - \alpha(G) = 12+1 - 2= 11$$

$$\gamma(G) \geq \omega(G) =8$$

$$8 \leq \gamma(G) \leq 11$$

II-Validation d'une application critique de fonctionnement (12 points)

Code thread	Durée maximale	thread(s) précédent(s)
X_1	4	-
X_2	8	-
X_3	1	-
X_4	1	X_3
X_5	6	X_1
X_6	3	X_1
X_7	5	X_2
X_8	3	X_5, X_6 , X_7

Code thread	Durée maximale	thread(s) précédent(s)
X_9	1	X_4
X_{10}	2	X_9
X_{11}	2	X_8
X_{12}	5	X_{10}, X_{11}

1-Proposer un modèle de graphe du cycle

Le modèle retenu est un graphe orienté

G(S, A)

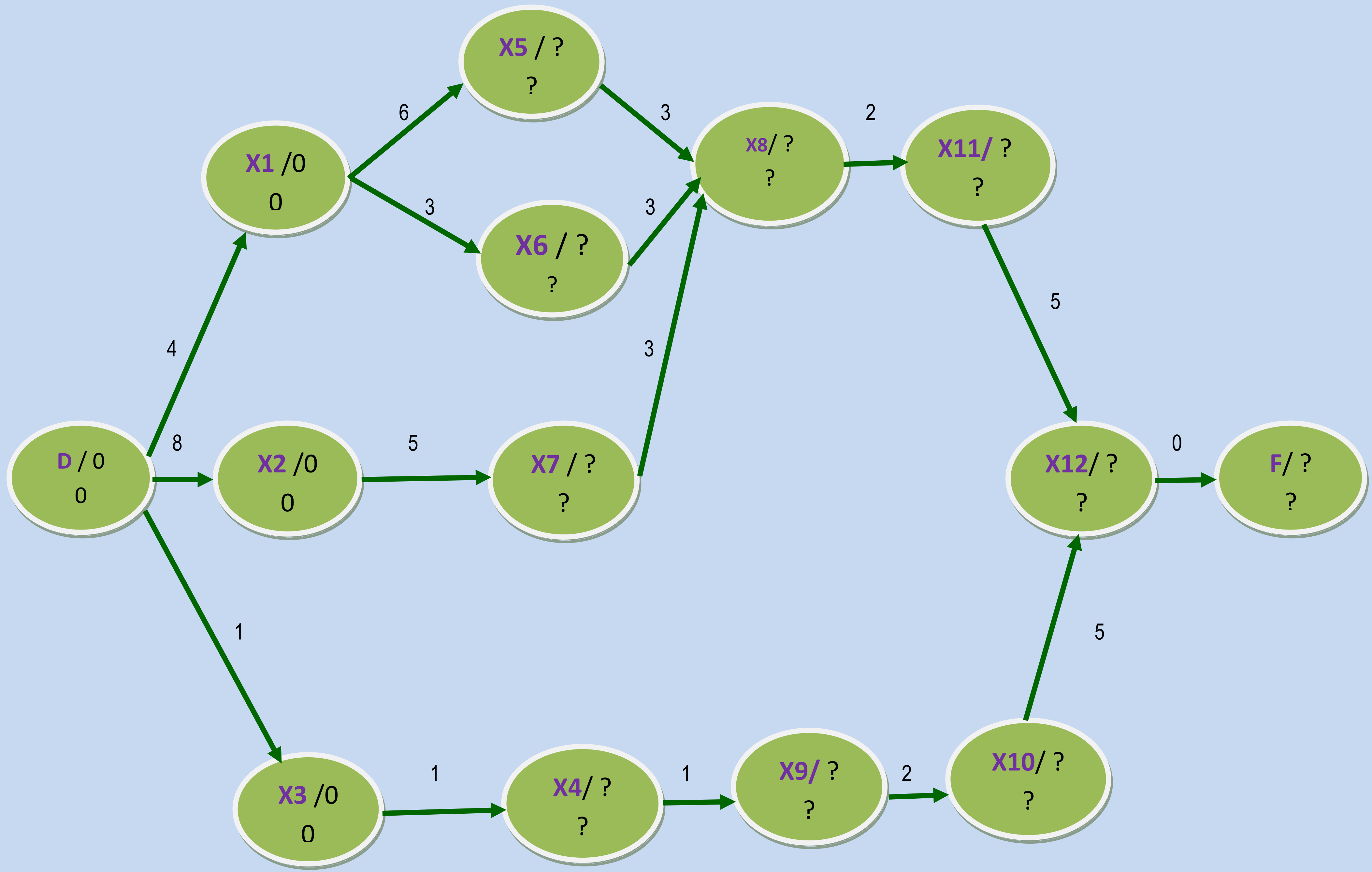
Où:

S= { **D**, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12,**F** }

A= { (D, X1), (D, X2), (D, X3), (X1, X5), (X1 ,X6), (X2, X7),
(X3, X4), (X5, X8), (X6, X8), (X7, X8), (X4, X9), (X8, X11),
(X9, X10), (X10, X12), (X11, X12), (X12, F) }

On choisit la convention suivante pour représenter les sommets du modèle de graphe.

D’où le modèle de graphe suivant :

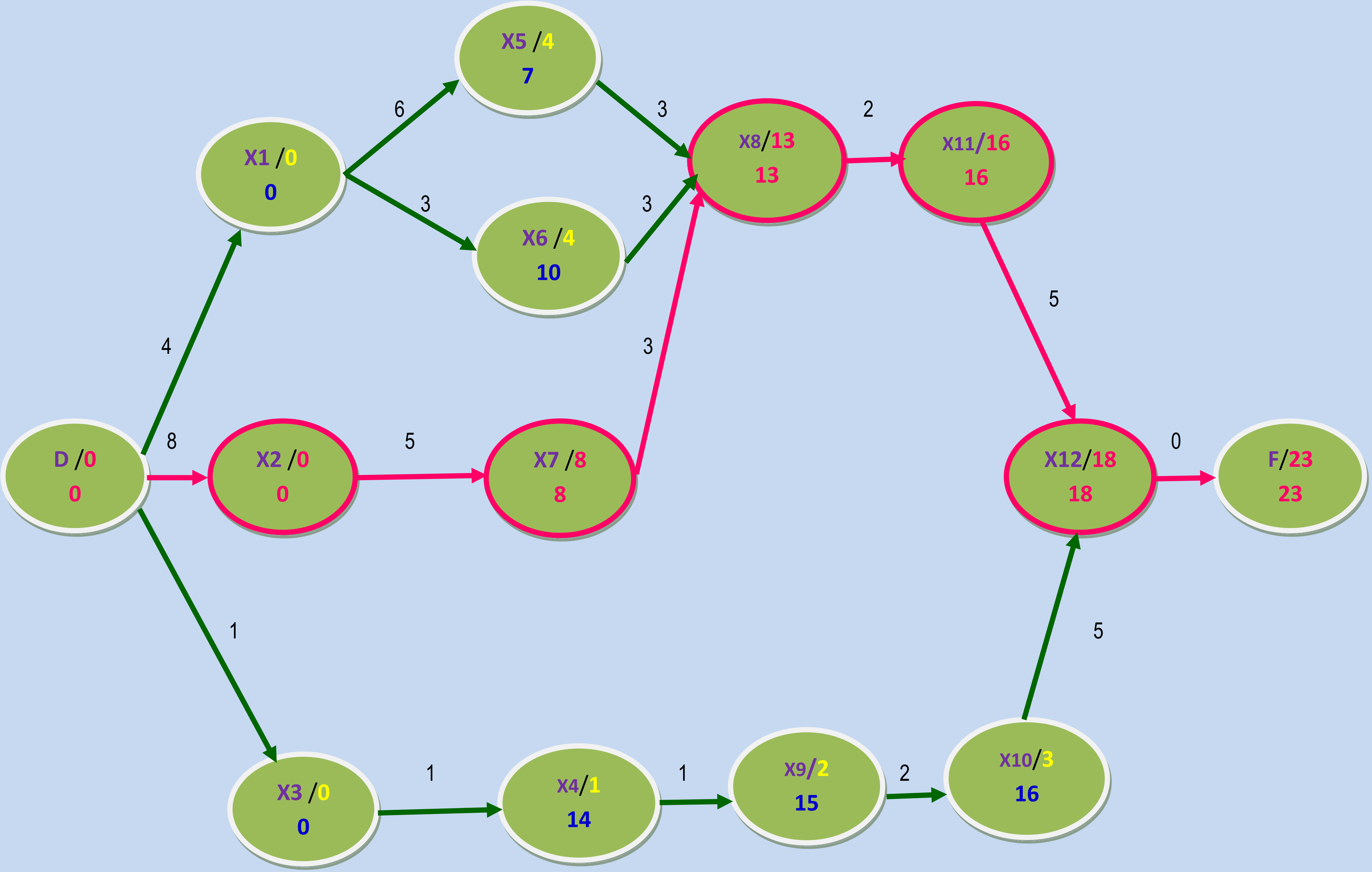

D / 0
0
X1 /0
0
X2 /0
0
X3 /0
0
X4/ ?
?
X5 / ?
?
X6 / ?
?
X7 / ?
?
X8/ ?
?
X9/ ?
?
X10/ ?
?
X11/ ?
?
X12/ ?
?
F/ ?
?
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0

Le modèle sera complet dès lors que sera calculé pour chaque tâche:

-sa date de départ au **plus tôt**

-et sa date de départ au **plus tard**.

A titre indicatif, le modèle complet sera le suivant :



X5 /4
7
X1 /0
0
X6 /4
10
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2
15
X10/3
16
6
3
3
3
2
4
8
5
3
5
0
1
1
1
2
5

2- Montrer que le problème de calcul de la date d'activation **au plus tôt** d'un thread **T** se ramène au problème de recherche du **chemin le plus long**

Pour préserver la relation de précédence on doit imposer qu'un thread T :

-ne peut être exécutée

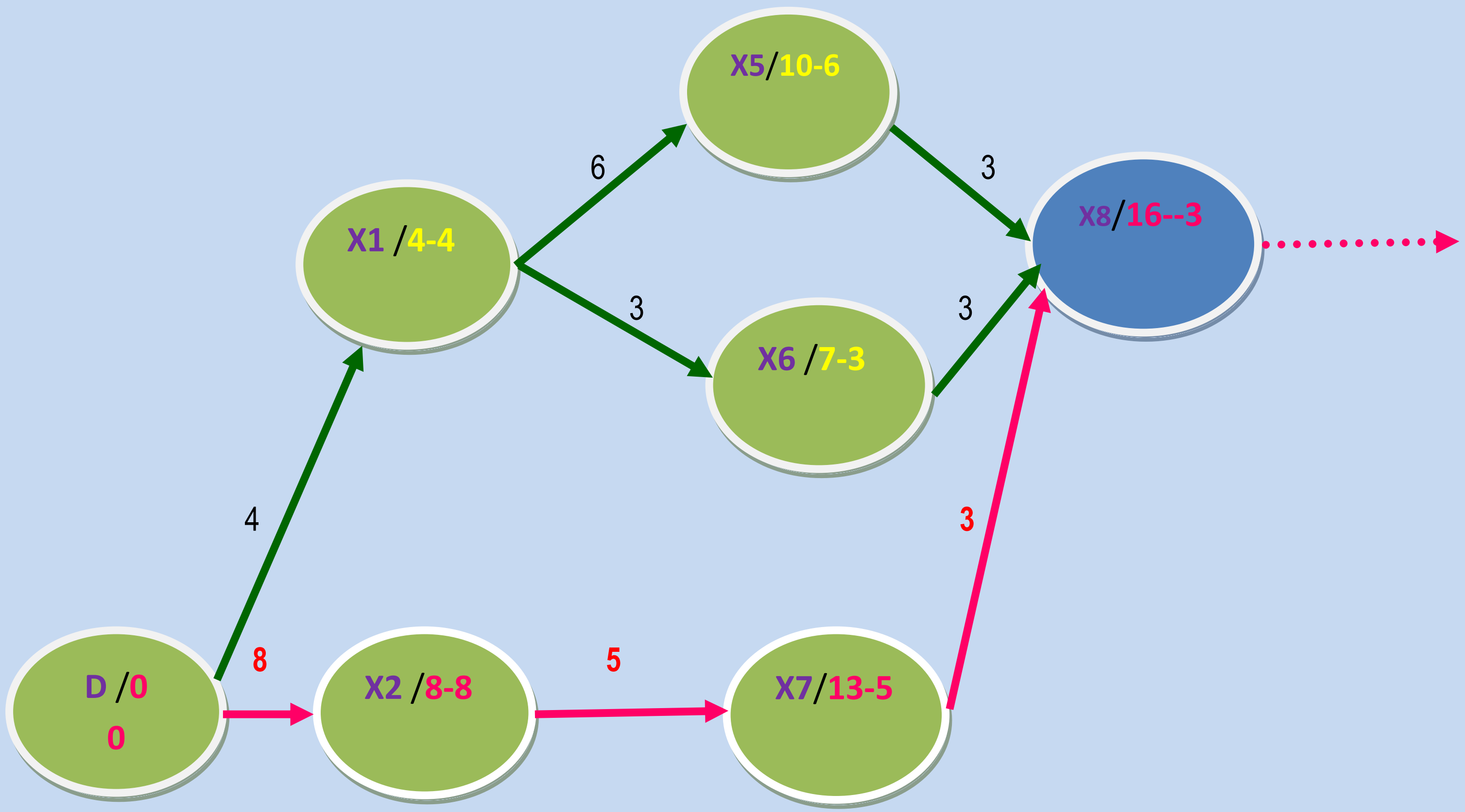
-qu'une fois exécutés **tous** les threads qui le précèdent.

Donc, la détermination de la date au plus tôt d'un thread **T** implique le calcul :

- de la longueur $\mathbf{L_T}$ du chemin le plus long
- entre les sommets **DEBUT** et **T**

Puisqu'on calcule la date de **départ** du thread T (son exécution n'ayant pas démarré) il convient de **soustraire :**

- de la longueur $\mathbf{L_T}$ du chemin le plus long,
- la durée $\mathbf{t_T}$ du thread **T**.



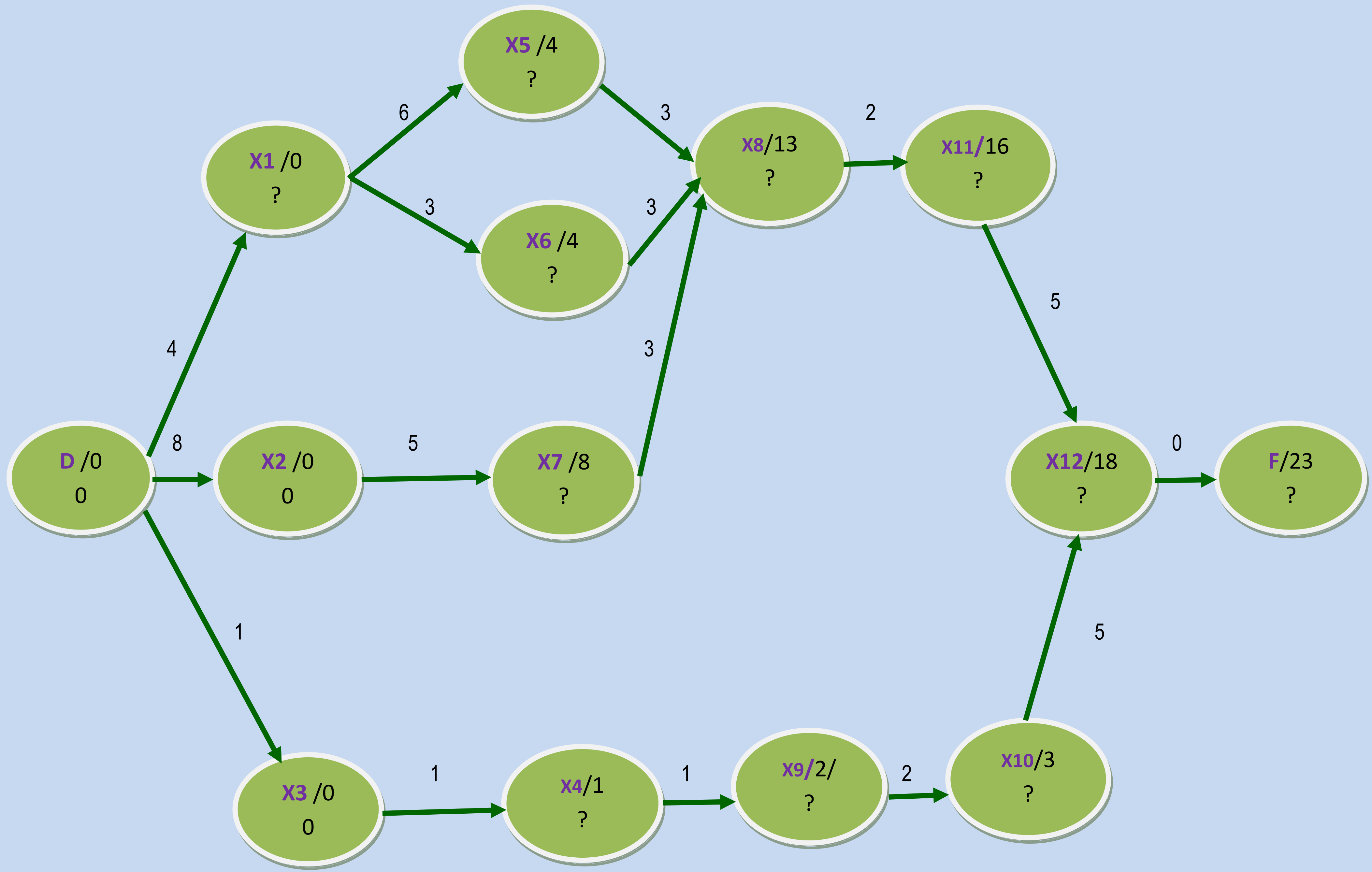
$$\text{Dtôt}(X8) = L_8 - t_8$$

$$= 16 - 3 = 13$$

D'où la formule suivante de calcul de la date au plus tôt d'un thread T:

$$\text{Dtôt(T)} = \mathbf{L_T} - \mathbf{t_T}$$

Appliquée au modèle de graphe initial, on obtient le modèle suivant plus complet :



D /0
0
X1 /0
?
X2 /0
0
X3 /0
0
X4/1
?
X5 /4
?
X6 /4
?
X7 /8
?
X8/13
?
X9/2/
?
X10/3
?
X11/16
?
X12/18
?
F/23
?
4
8
1
6
3
5
3
3
3
2
5
5
1
1
2
0

3-Déduire que ce qui précède que le problème d'optimisation de la durée du cycle peut être formulé en termes de recherche de chemin optimal entre le sommet **Début** et le sommet **Fin**.

En fixant à **0** la date de départ au plus tôt de la tâche DEBUT, la **date de départ au plus tôt** de la tâche FIN représente la durée optimale du cycle.

D'après 2) la date au plus tôt de la tâche FIN passe par la recherche du chemin le plus long partant du sommet DEBUT et atteignant le sommet FIN.

En appliquant la formule donnée en 2) on a :

$$\text{Dtôt(FIN)} = \mathbf{L_{FIN} - t_{FIN}}$$

Comme :

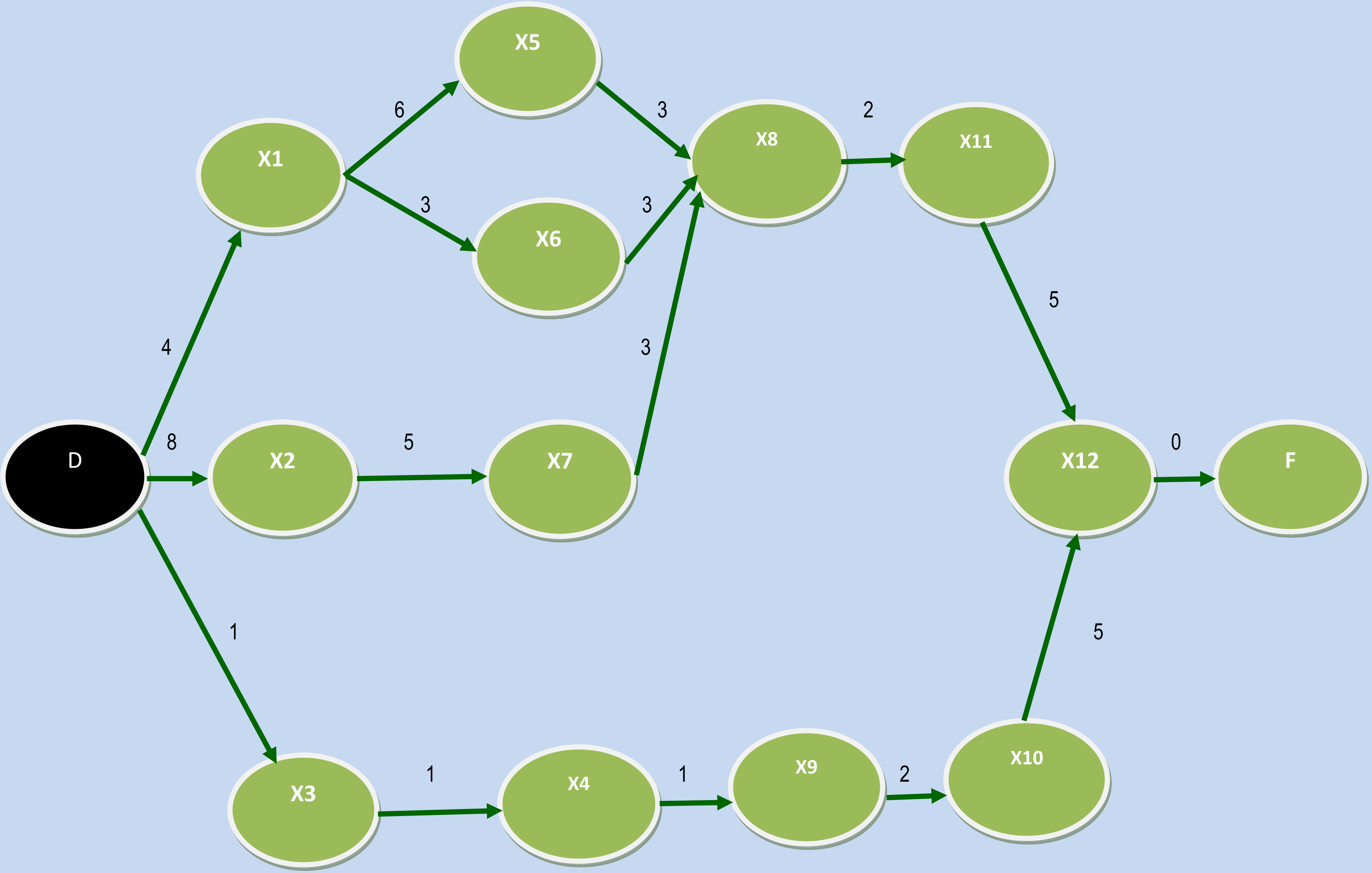
$$\mathbf{L_{FIN} = 23} \qquad \mathbf{t_{FIN}} = 0$$

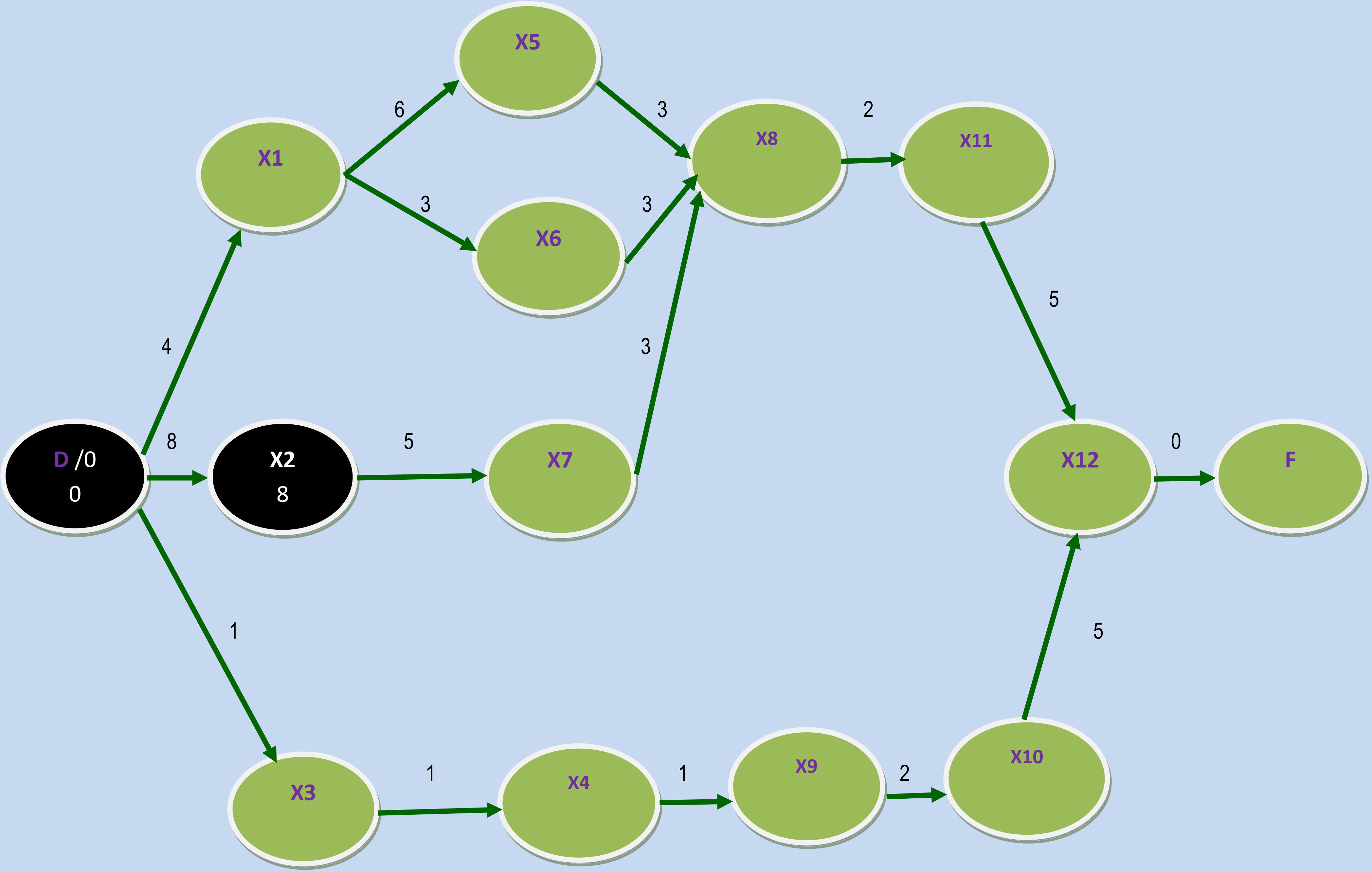
On a : $\text{Dtôt(FIN)} = 23$

4. Compte tenu de ce qui précède, proposer une solution pour calculer les dates au **plus tôt** d’activation des threads et la durée optimale du cycle d’exécution

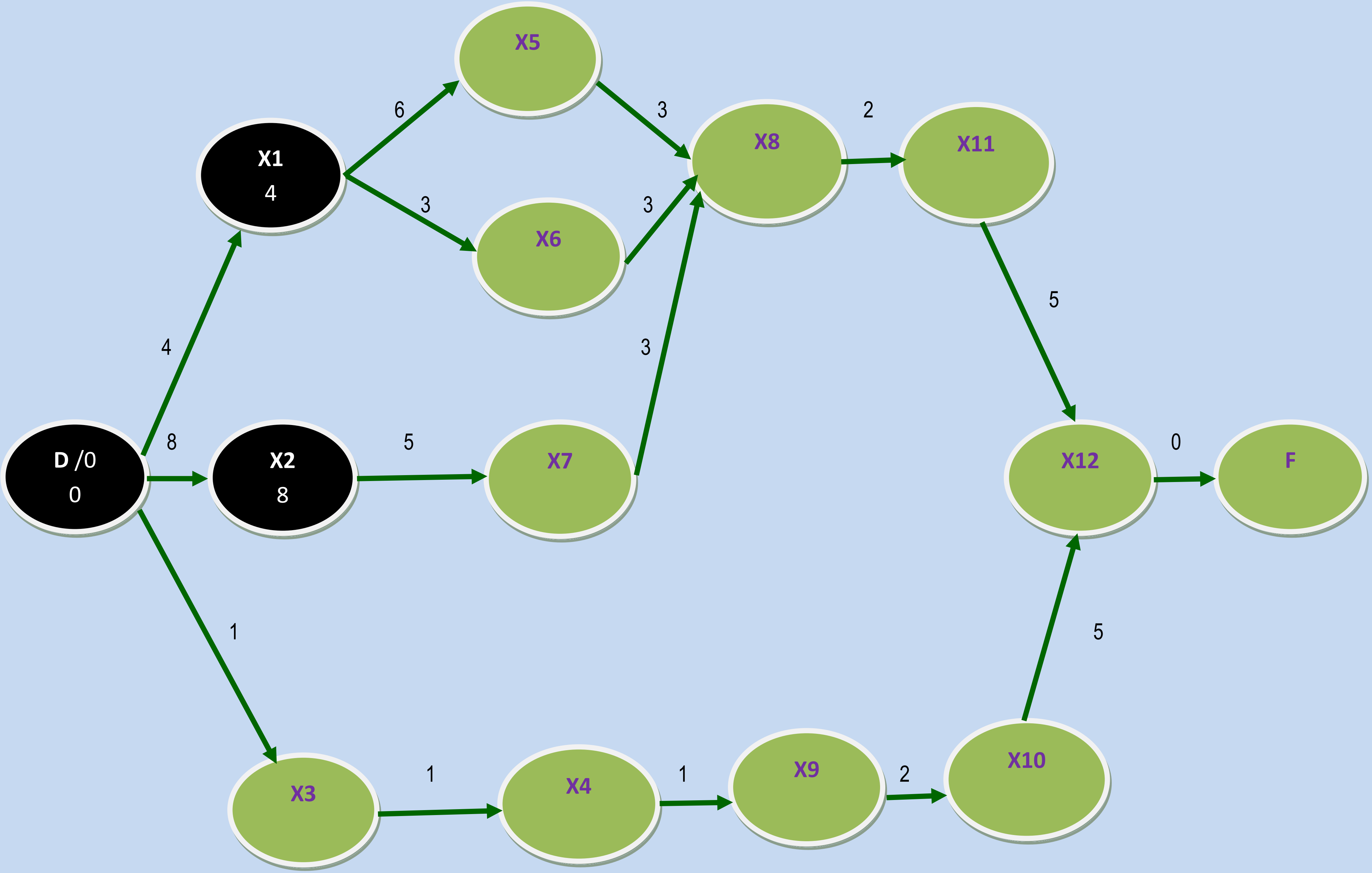
La procédure s’appuie sur la recherche dans le modèle de graphe le **chemin le plus long** entre les sommets.

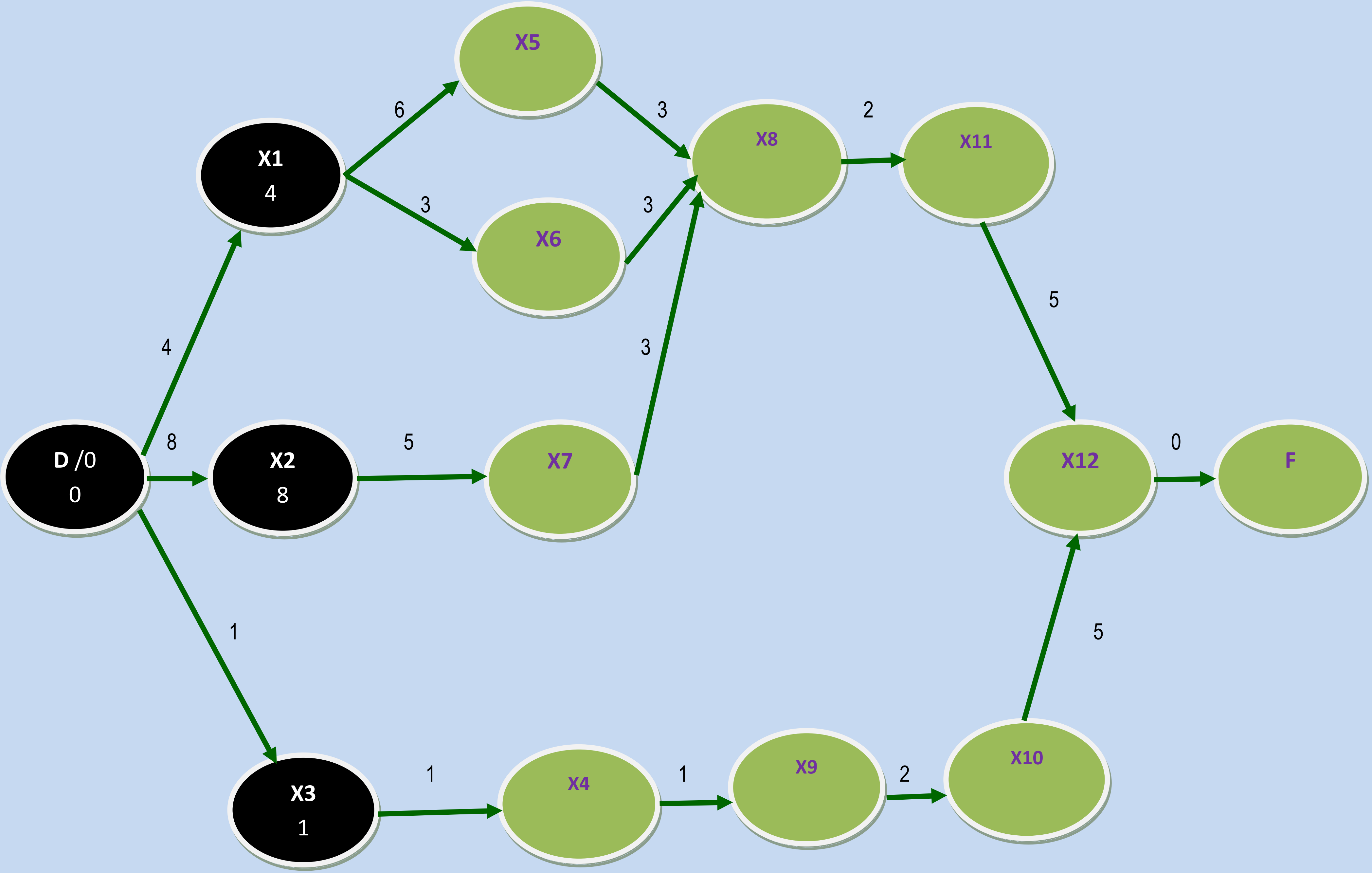
On peut appliquer directement la **procédure de Bellman** (voir cours).

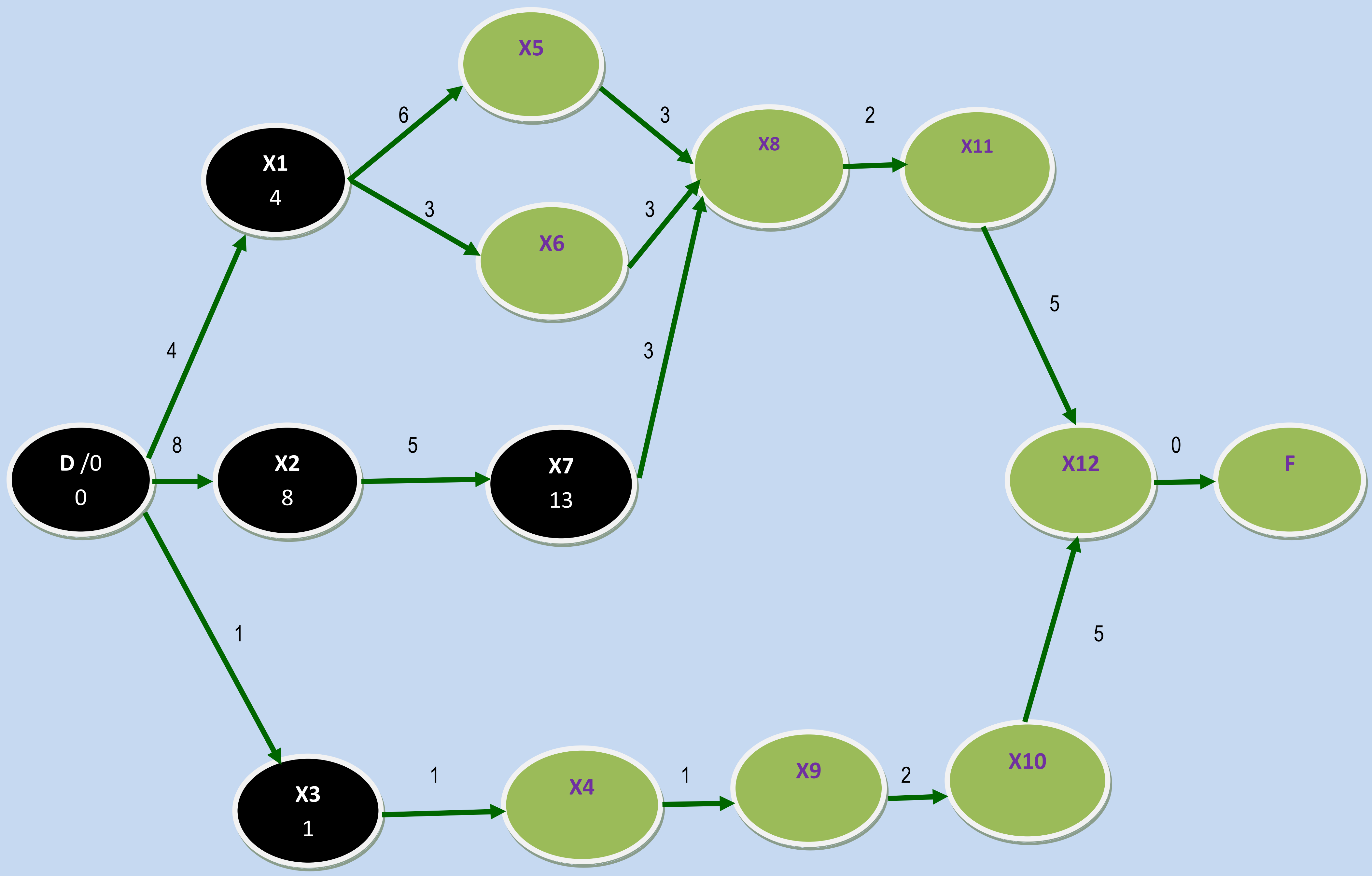

X5
6
3
X8
2
X11
X1
3
X6
3
5
4
3
D
8
X2
5
X7
X12
0
F
1
5
1
X3
X4
1
X9
2
X10

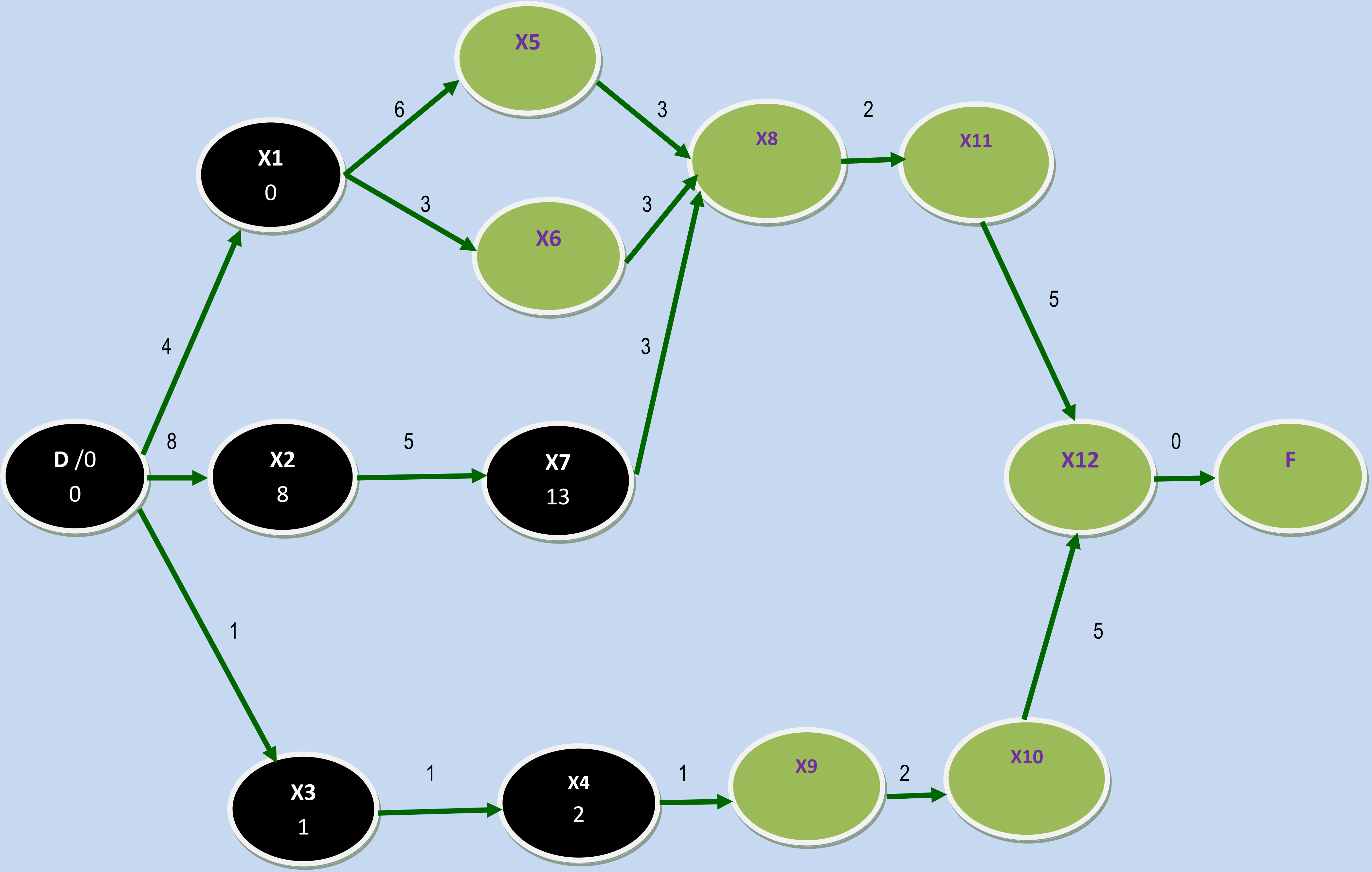


D /0
0
X1
X2
8
X3
X4
X5
X6
X7
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
3
3
3
2
5
0
5
1
1
2

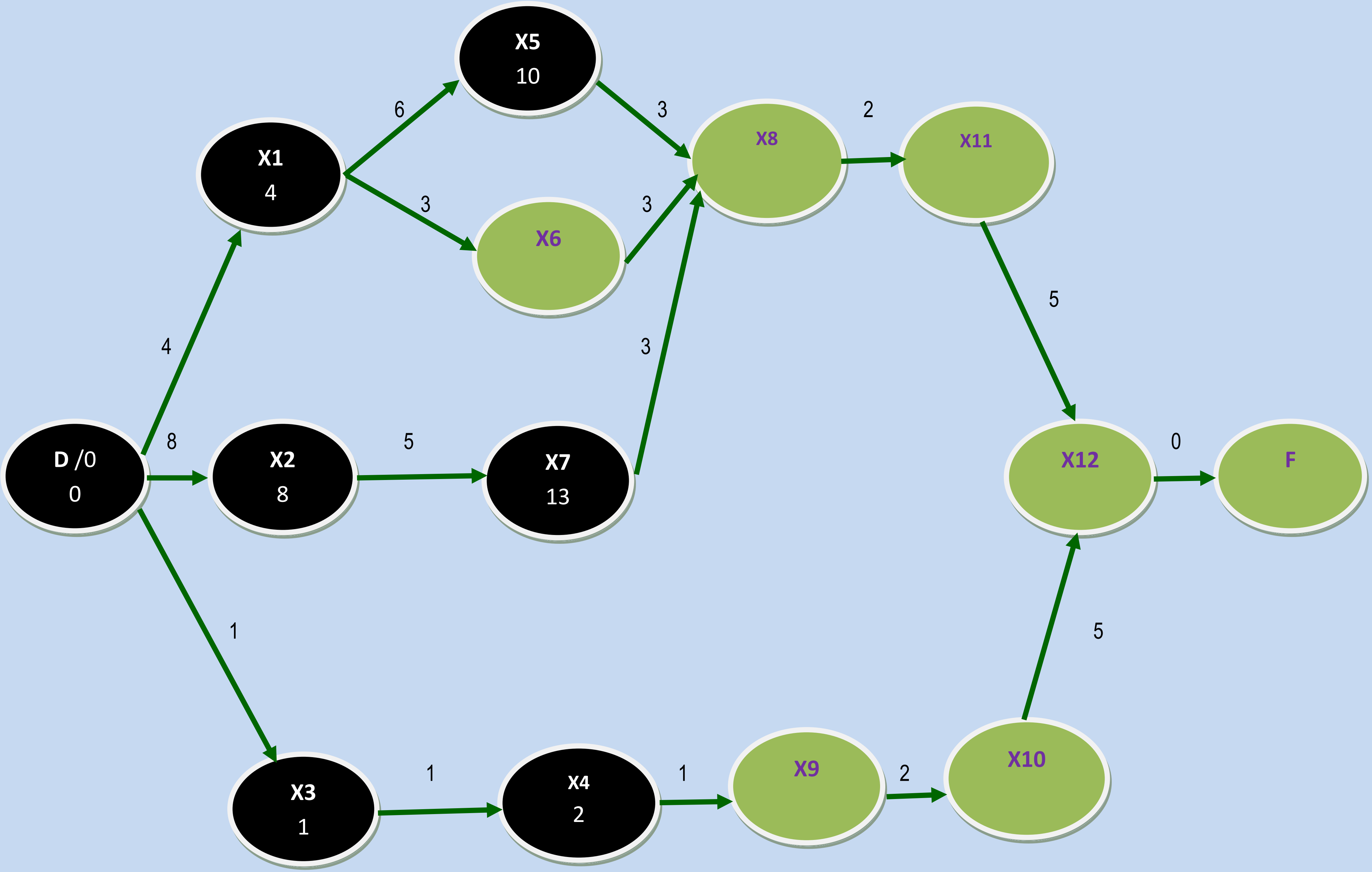

D /0
0
X1
4
X2
8
X3
X4
X5
X6
X7
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
3
3
3
2
5
1
1
2
5
0

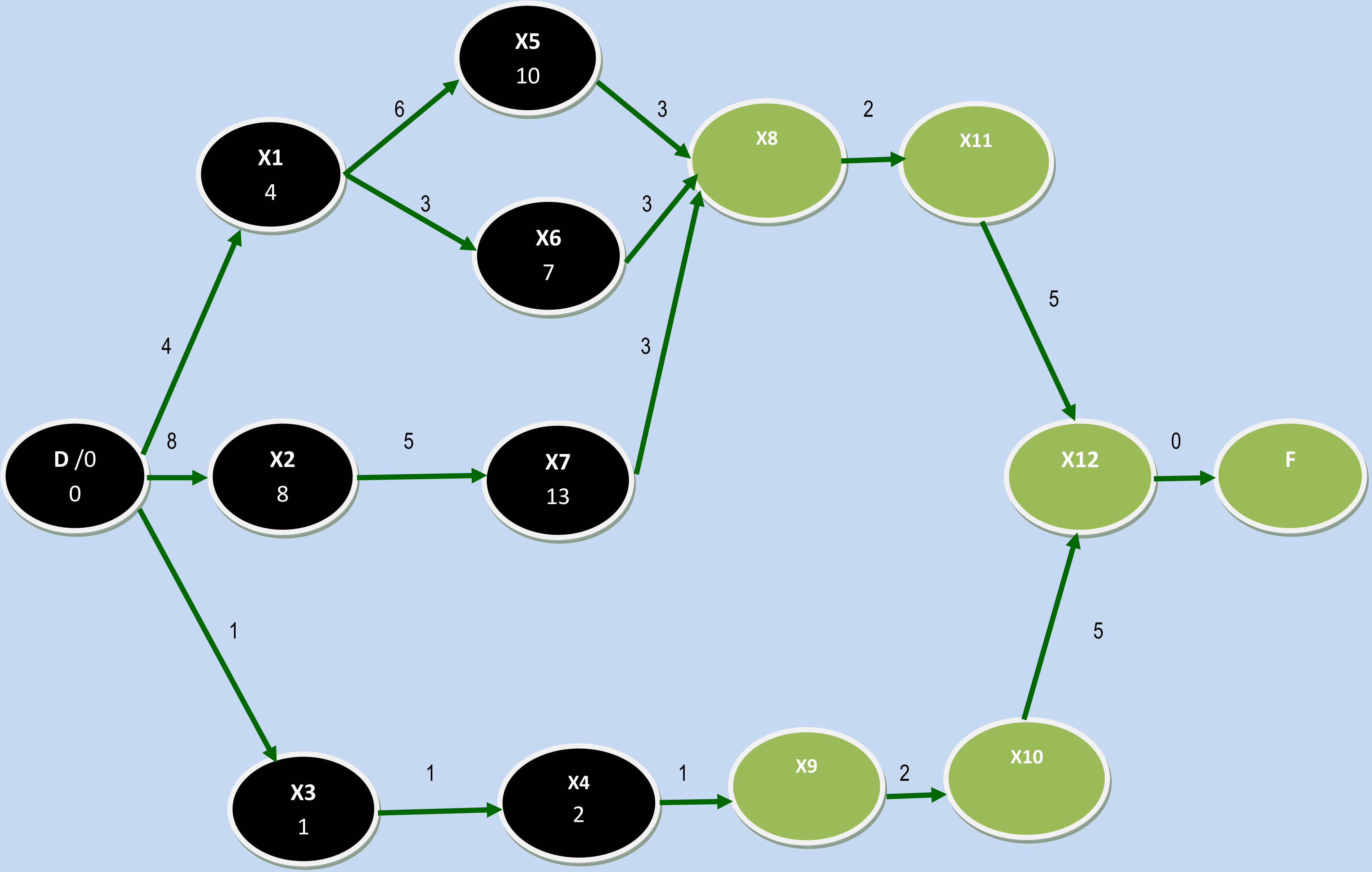

D /0
0
X1
4
X2
8
X3
1
X5
X6
X7
X4
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0


D /0
0
X1
4
X2
8
X3
1
X4
X5
X6
X7
13
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0

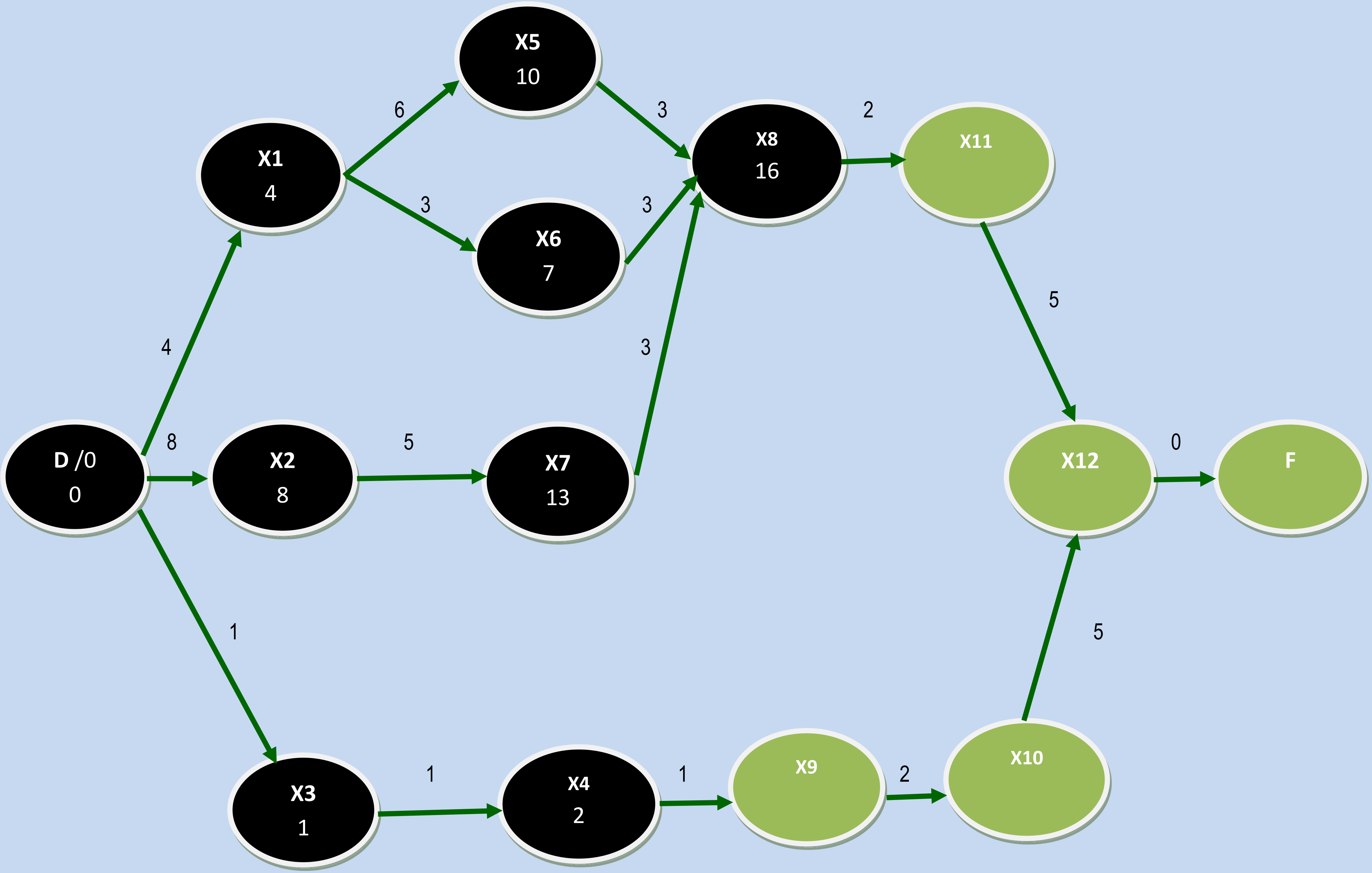


D /0
0
X1
0
X2
8
X3
1
X4
2
X5
X6
X7
13
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
3
3
3
2
5
5
1
1
2
0

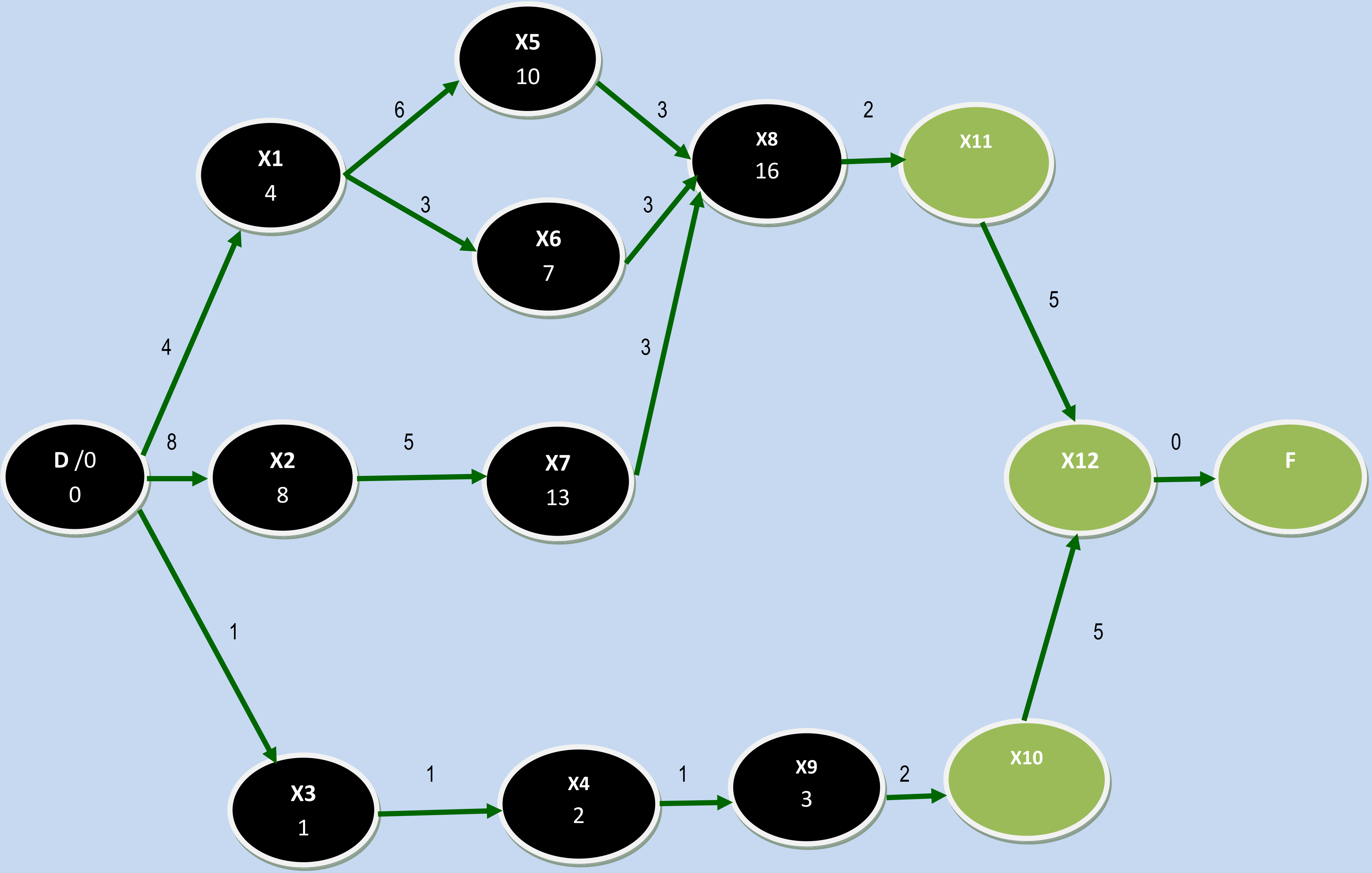

D /0
0
X1
4
X2
8
X3
1
X4
2
X5
10
X6
X7
13
X8
X9
X10
X11
X12
F
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0

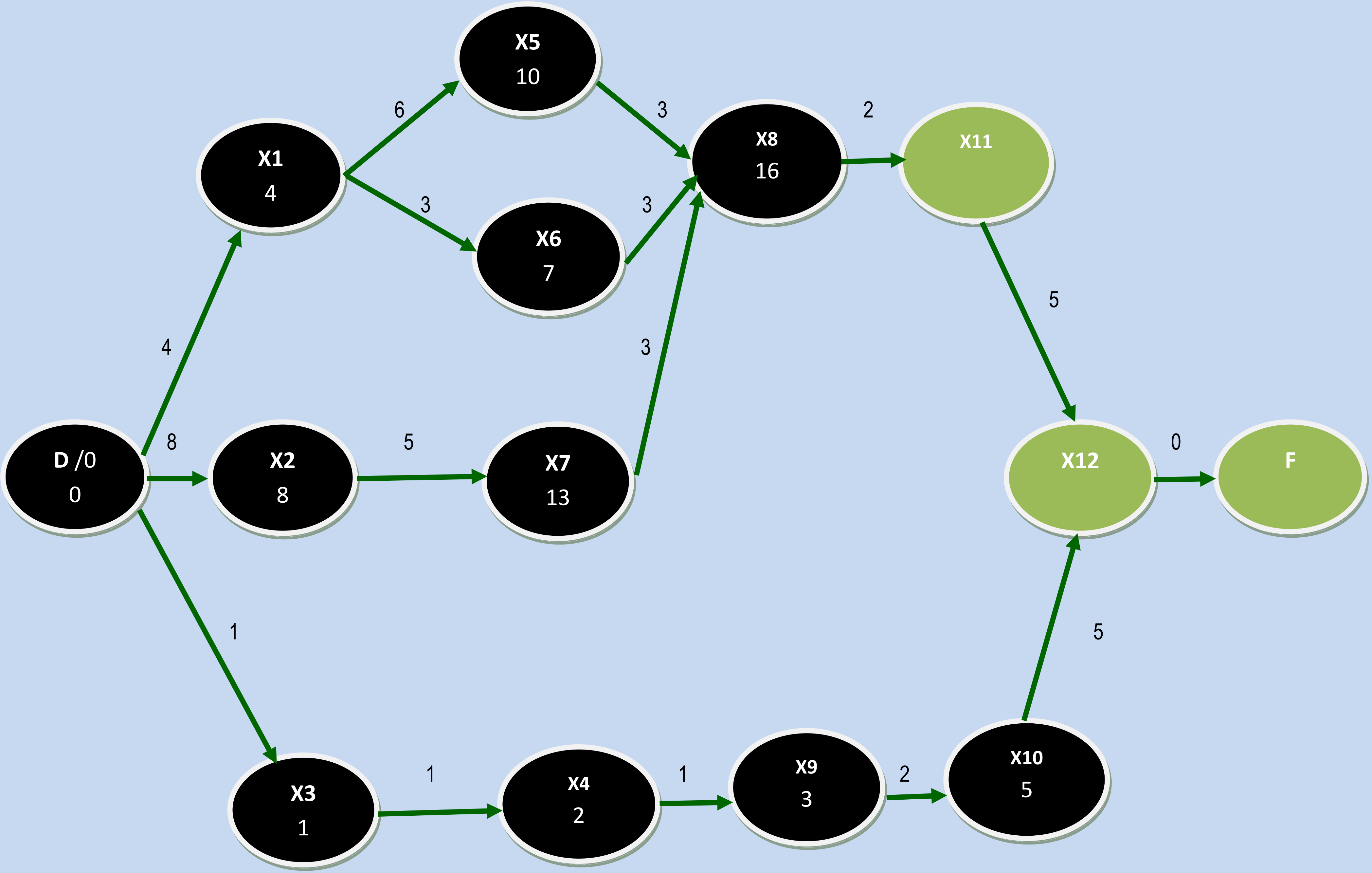


X5
10
X1
4
X6
7
X8
X11
D /0
0
X2
8
X7
13
X12
F
X3
1
X4
2
X9
X10
6
3
3
3
2
4
3
5
8
5
0
1
1
1
2
5

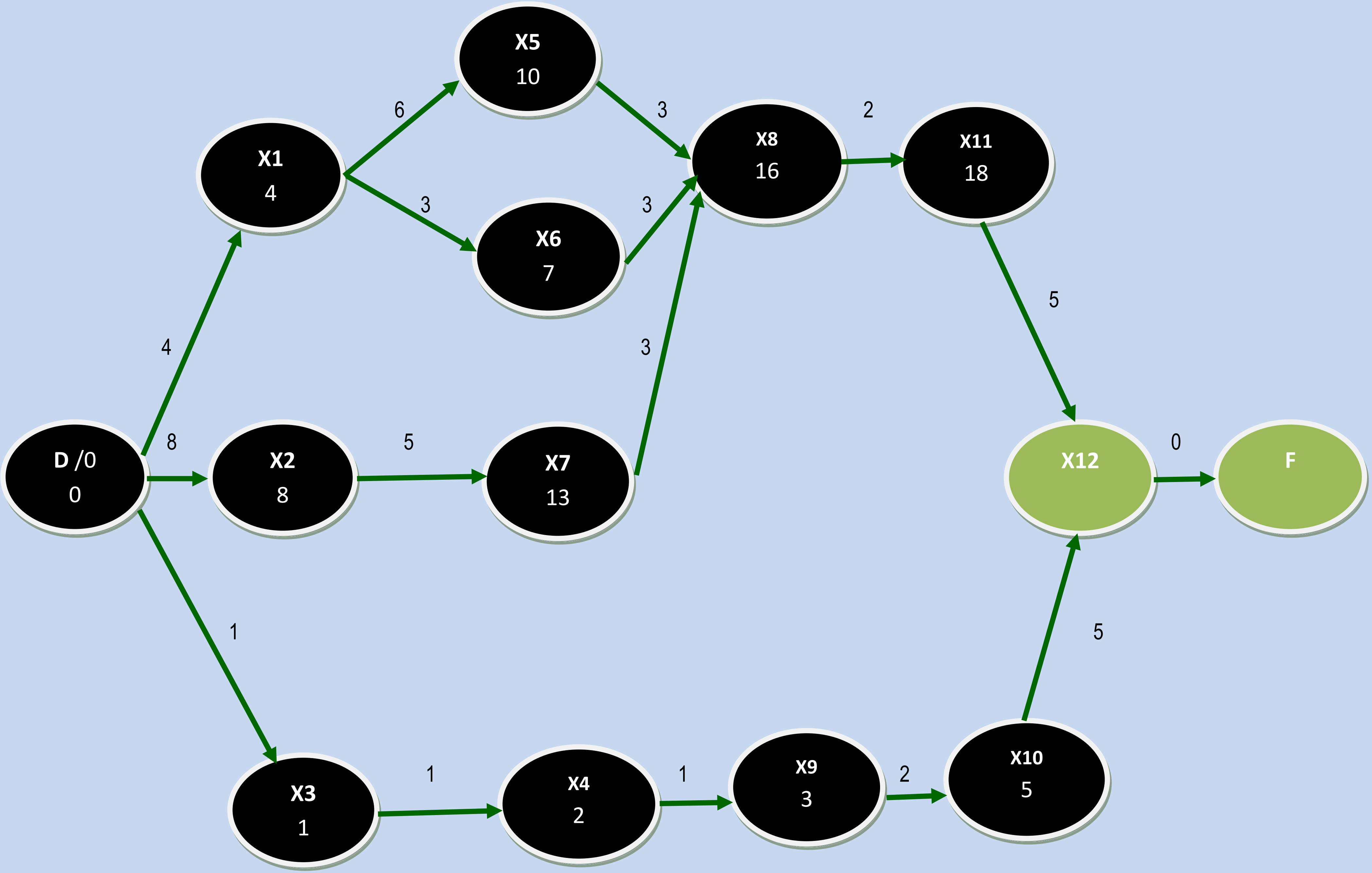


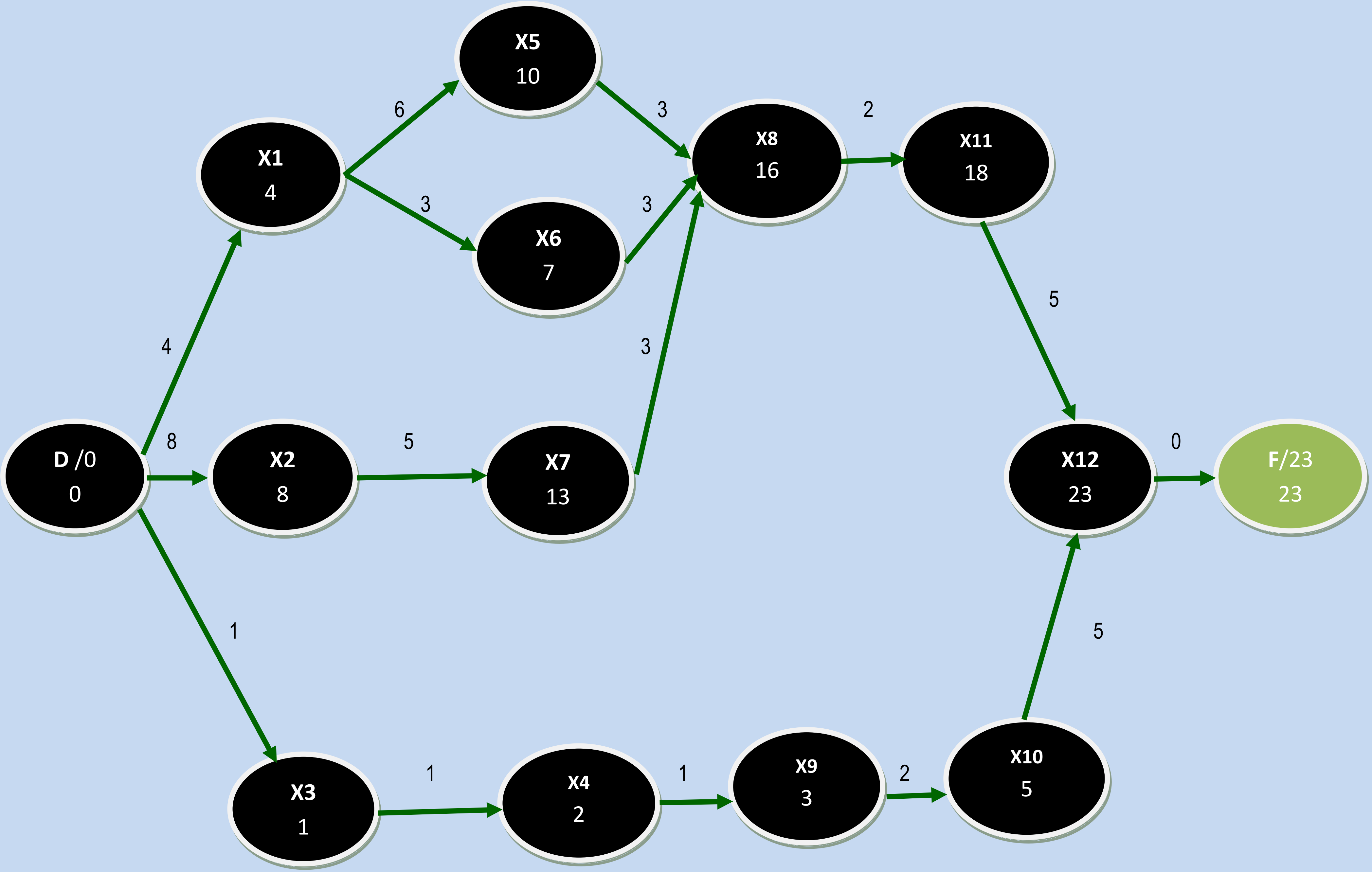
D /0
0
X1
4
X2
8
X3
1
X4
2
X5
10
X6
7
X7
13
X8
16
X9
X10
X11
X12
F
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0

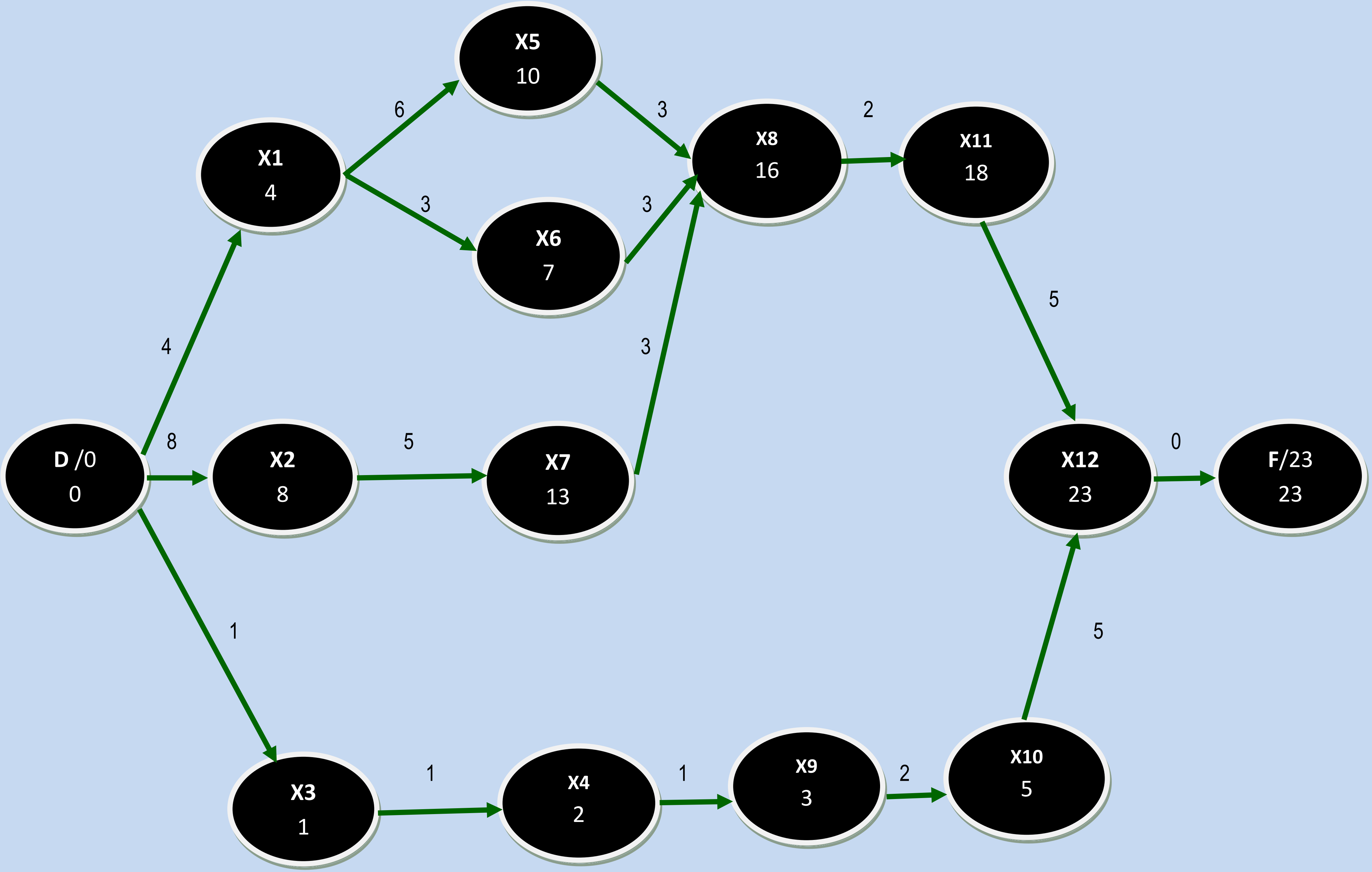

X5
10
X1
4
X6
7
X8
16
X11
D /0
0
X2
8
X7
13
X12
F
X3
1
X4
2
X9
3
X10
6
3
3
3
2
4
8
5
3
5
1
1
1
2
5
0

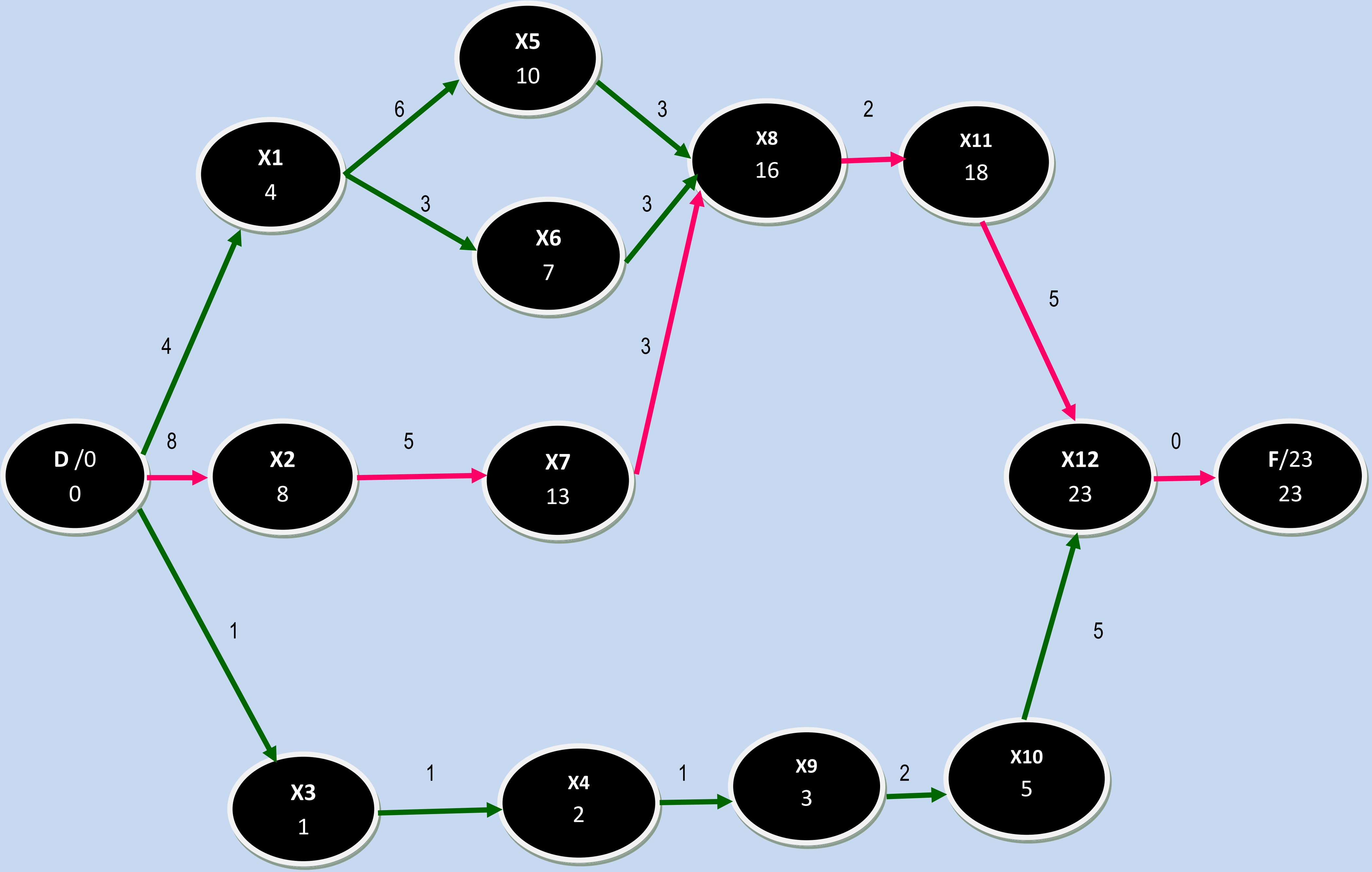


X5
10
6
3
X8
16
2
X11
X1
4
3
X6
7
3
5
4
3
8
D /0
0
X2
8
5
X7
13
X12
0
F
1
5
X3
1
1
X4
2
1
X9
3
2
X10
5

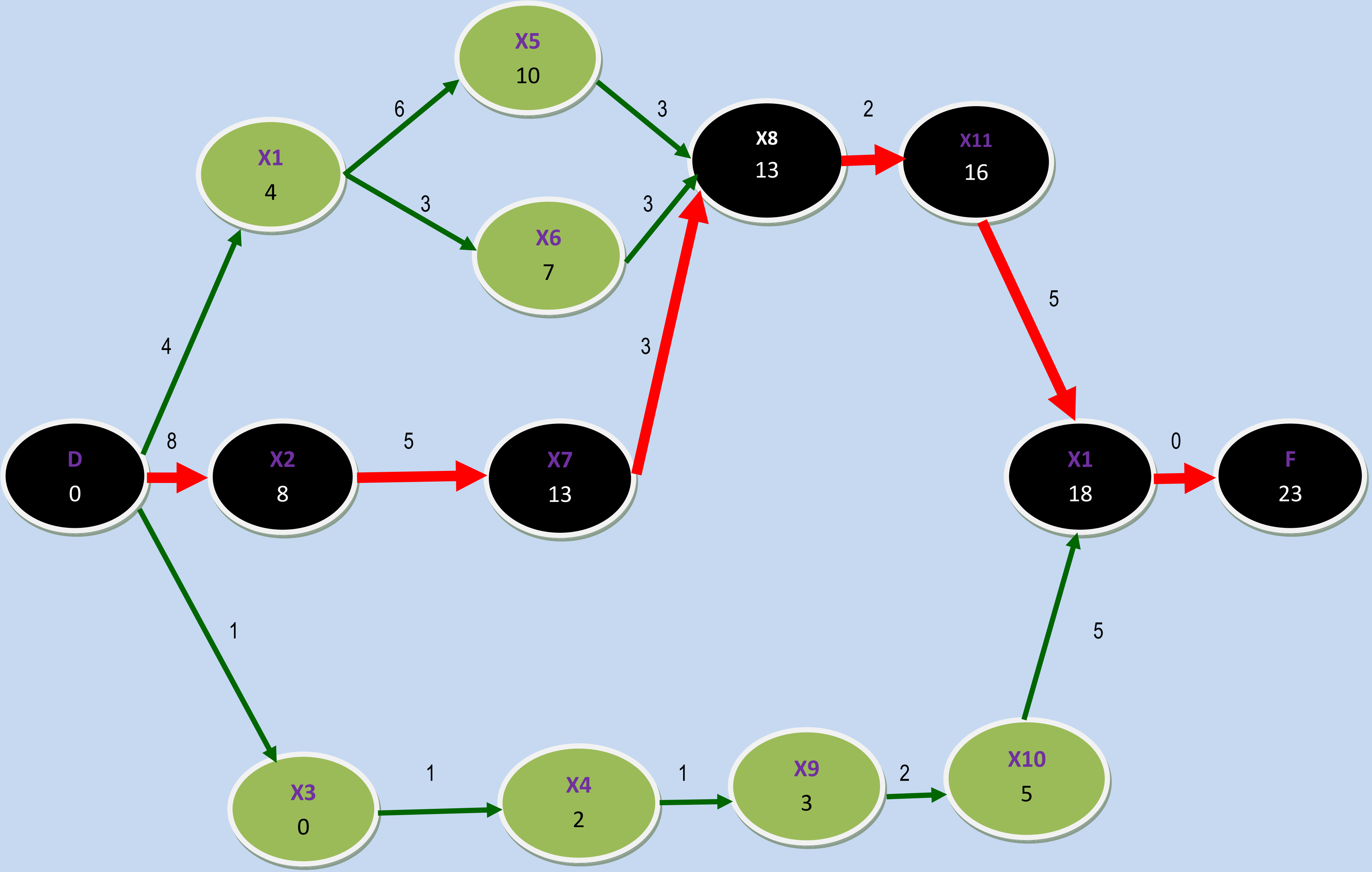

D /0
0
X1
4
X2
8
X3
1
X4
2
X5
10
X6
7
X7
13
X8
16
X9
3
X10
5
X11
18
X12
F
4
8
1
6
3
5
3
3
3
2
1
1
2
5
5
0


D /0
0
X1
4
X2
8
X3
1
X4
2
X5
10
X6
7
X7
13
X8
16
X9
3
X10
5
X11
18
X12
23
F/23
23
4
8
1
6
3
5
3
3
3
2
1
1
2
5
5
0

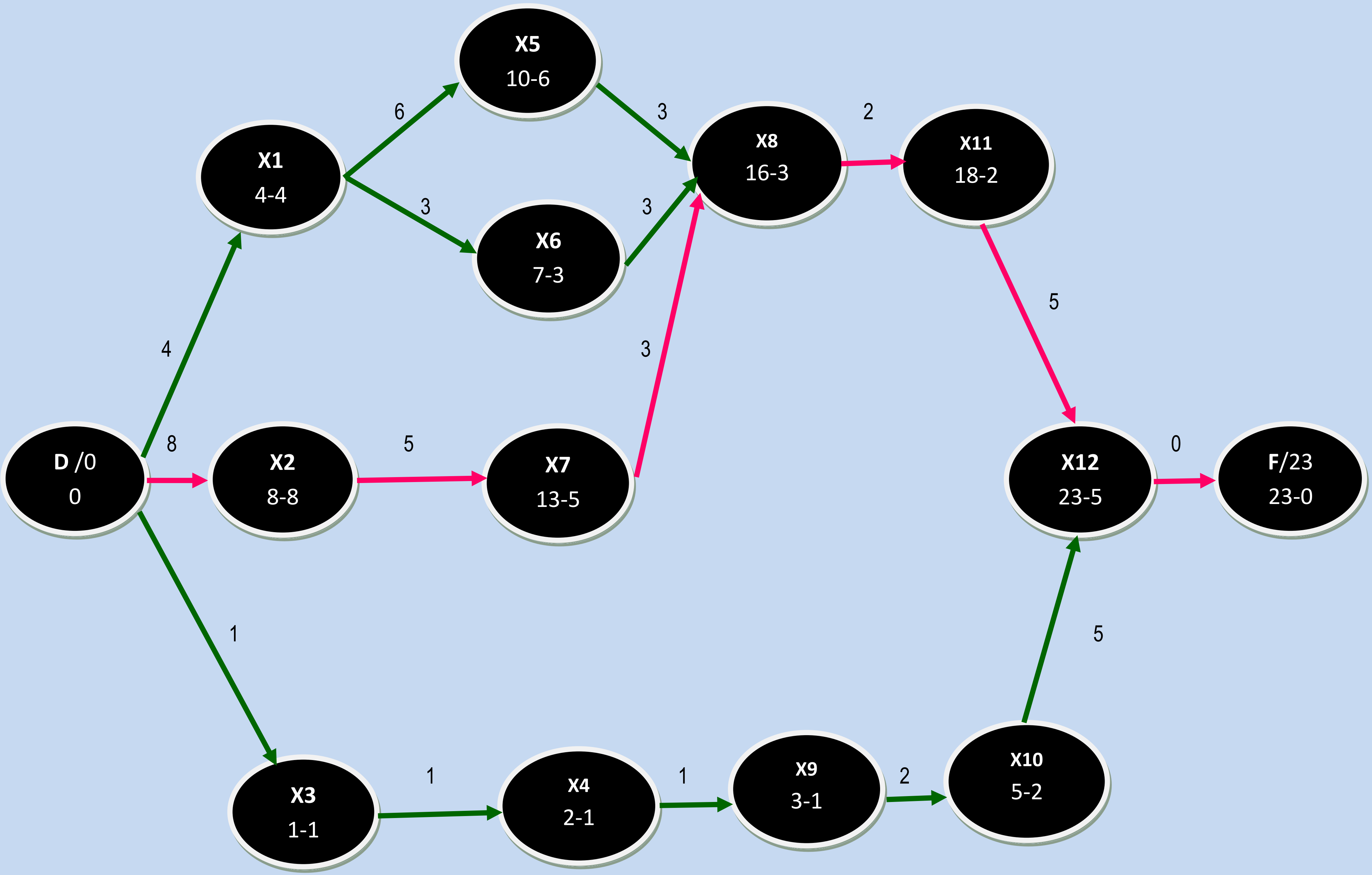

X5
10
X1
4
X6
7
X8
16
X11
18
D /0
0
X2
8
X7
13
X12
23
F/23
23
X3
1
X4
2
X9
3
X10
5
6
3
3
3
2
4
3
5
8
5
0
1
5
1
1
2


X5
10
X1
4
6
3
X8
16
2
X11
18
X6
7
3
3
4
3
5
D /0
0
8
X2
8
5
X7
13
X12
23
0
F/23
23
1
5
1
1
2
X3
1
X4
2
X9
3
X10
5

Résultat de l'algorithme de Bellman



Calcul des dates au plus tôt



5. Montrer que le problème de calcul des dates **au plus tard** passe par le problème de recherche :

- de chemin **le plus court**
- entre **chaque sommet** et le sommet **Fin**.

L’objectif est d’estimer la durée maximale :

- dont il est possible **retarder le démarrage** d’une tâche

- sans pour autant augmenter la **durée optimale** du cycle.

Pour cela, on se réfère à la **durée minimale** qui peut séparer :

- la date de démarrage (au plus tard) de la tâche,
- et la date du fin de cycle.

D'où le calcul pour chaque tâche **T** du **chemin le plus court** entre le sommet **Fin** et le sommet représentant **T.**

6-Proposer une solution pour calculer les dates au plus tard

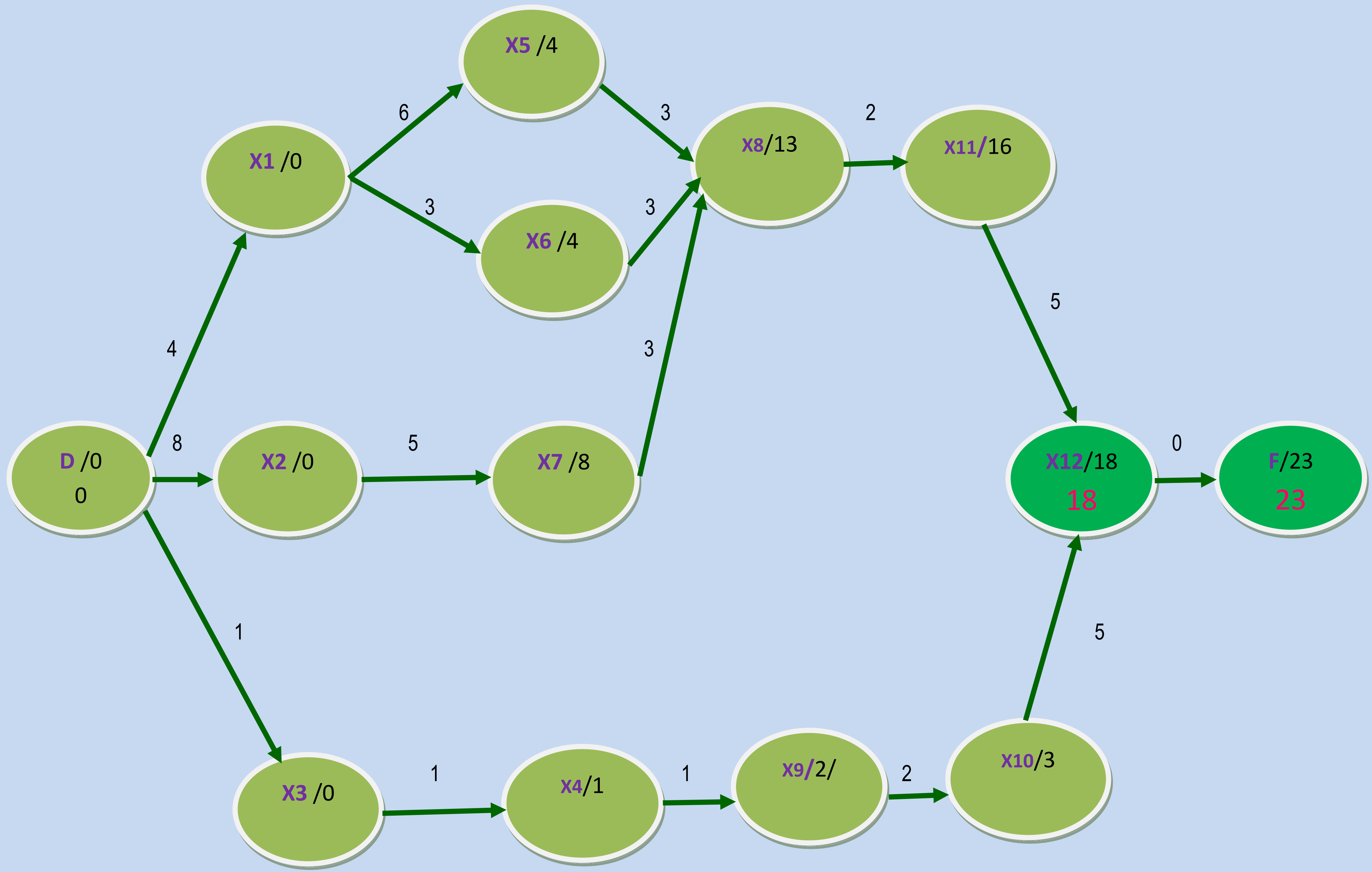
Notant $\mathbf{C_T}$, la longueur du plus court chemin entre :

- le thread **T**
- et la tâche **FIN**

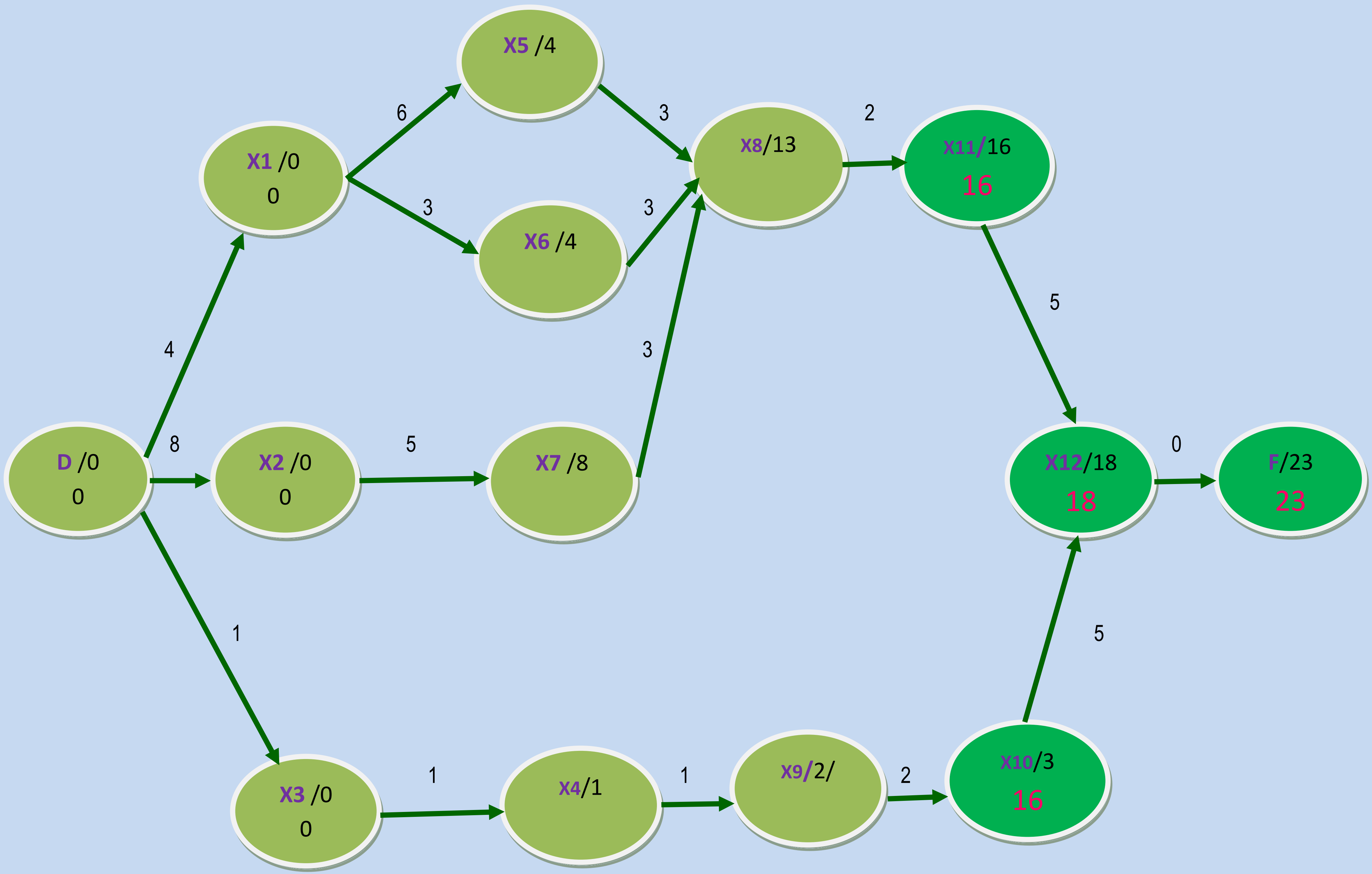
et $\mathbf{t_T}$ la durée du thread T

La date au plus tard est calculée à l'aide de la formule :

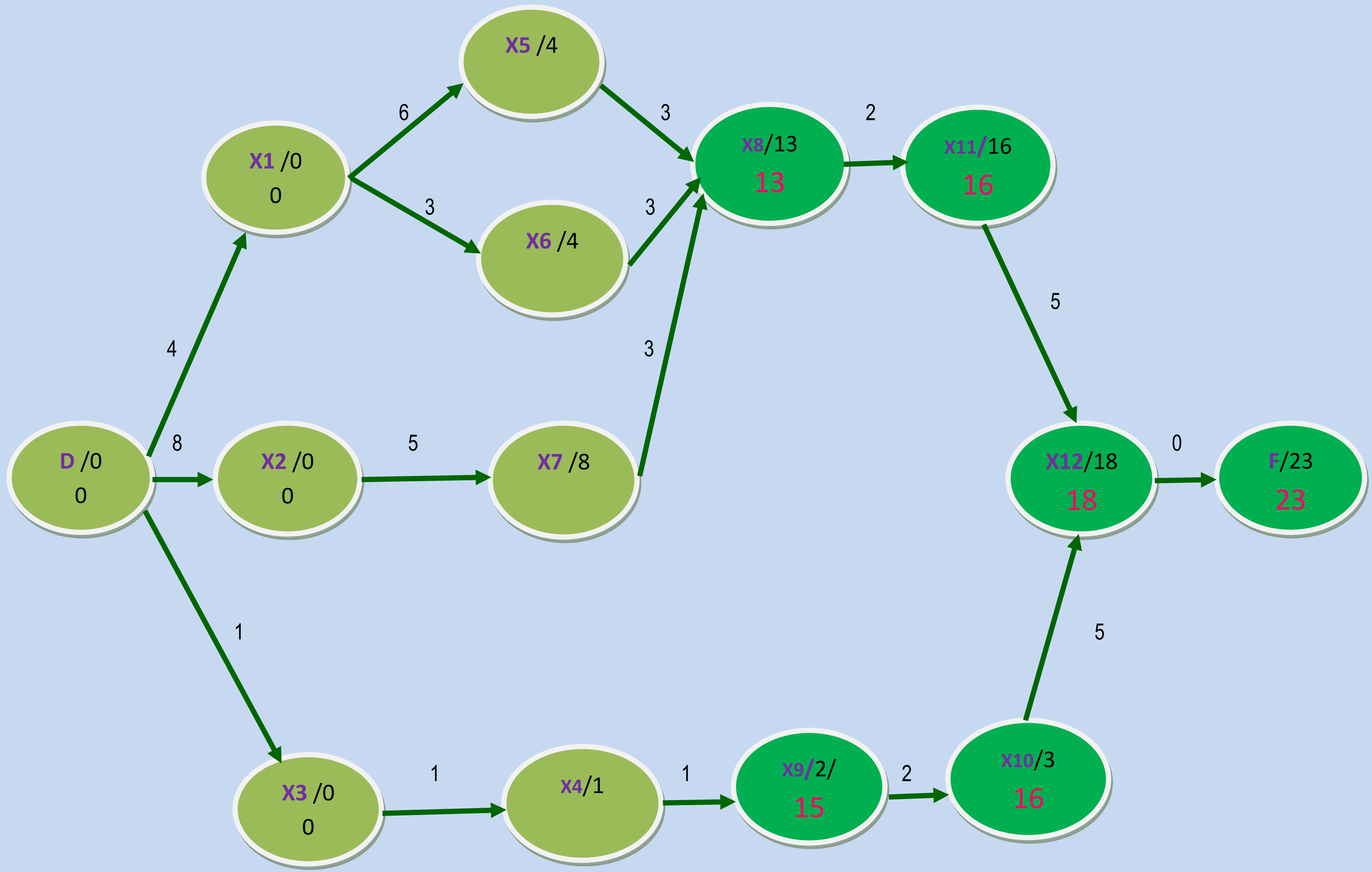
$$Dtard(T) = 23 - C_T - \mathbf{t_T}$$



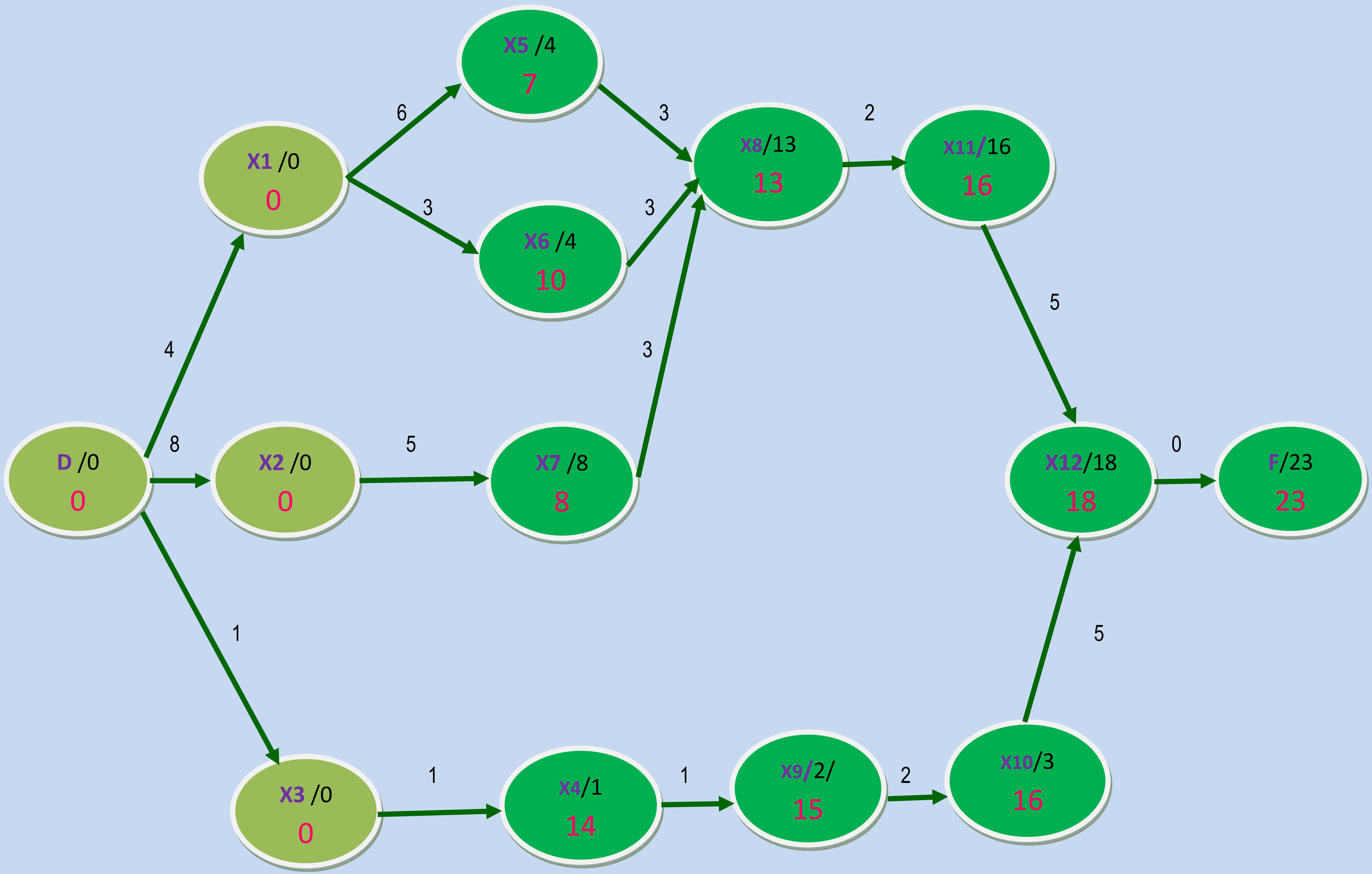
X5 /4
6
3
X8/13
2
X11/16
X1 /0
3
X6 /4
3
5
4
3
D /0
0
8
X2 /0
5
X7 /8
X12/18
18
0
F/23
23
1
5
X3 /0
1
X4/1
1
X9/2/
2
X10/3



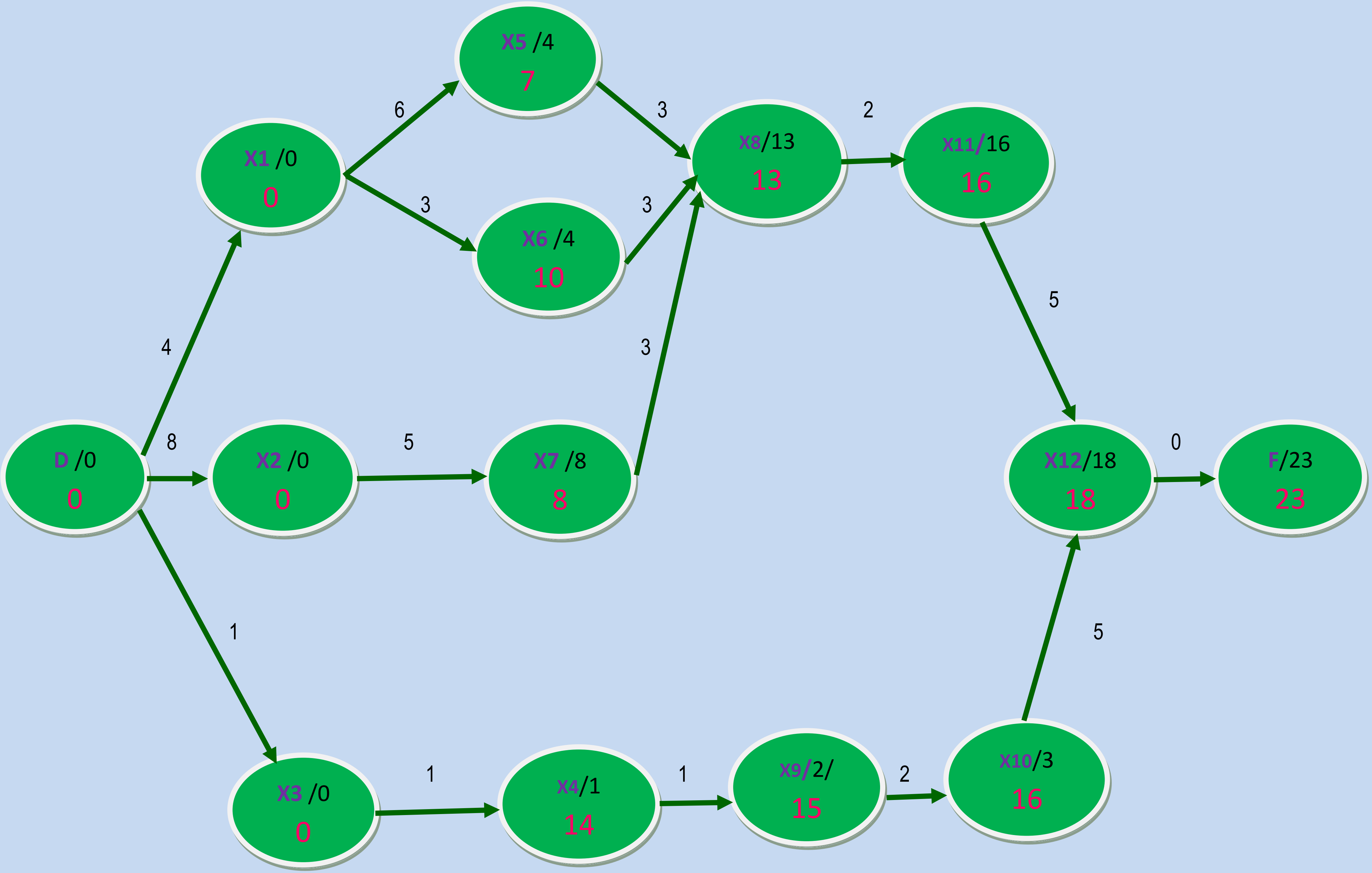
X5 /4
6
3
X8/13
2
X11/16
16
X1 /0
0
3
X6 /4
3
5
4
3
D /0
0
8
X2 /0
0
5
X7 /8
X12/18
18
0
F/23
23
1
5
X3 /0
0
1
X4/1
1
X9/2/
2
X10/3
16



D /0
0
X1 /0
0
X2 /0
0
X3 /0
0
X4/1
X5 /4
X6 /4
X7 /8
X8/13
13
X9/2/
15
X10/3
16
X11/16
16
X12/18
18
F/23
23
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0



D /0
0
X1 /0
0
X2 /0
0
X3 /0
0
X4/1
14
X5 /4
7
X6 /4
10
X7 /8
8
X8/13
13
X9/2/
15
X10/3
16
X11/16
16
X12/18
18
F/23
23
4
8
1
6
3
5
1
3
3
3
1
2
2
5
5
0



X5 /4
7
X1 /0
0
X6 /4
10
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2/
15
X10/3
16
6
3
3
3
2
4
3
5
8
5
0
1
5
1
1
2

7-Ecrire une procédure pour déterminer le **chemin critique**:

Le chemin critique est déterminé par le chemin le long duquel s’inscrivent les sommets représentant les **tâches critiques**.

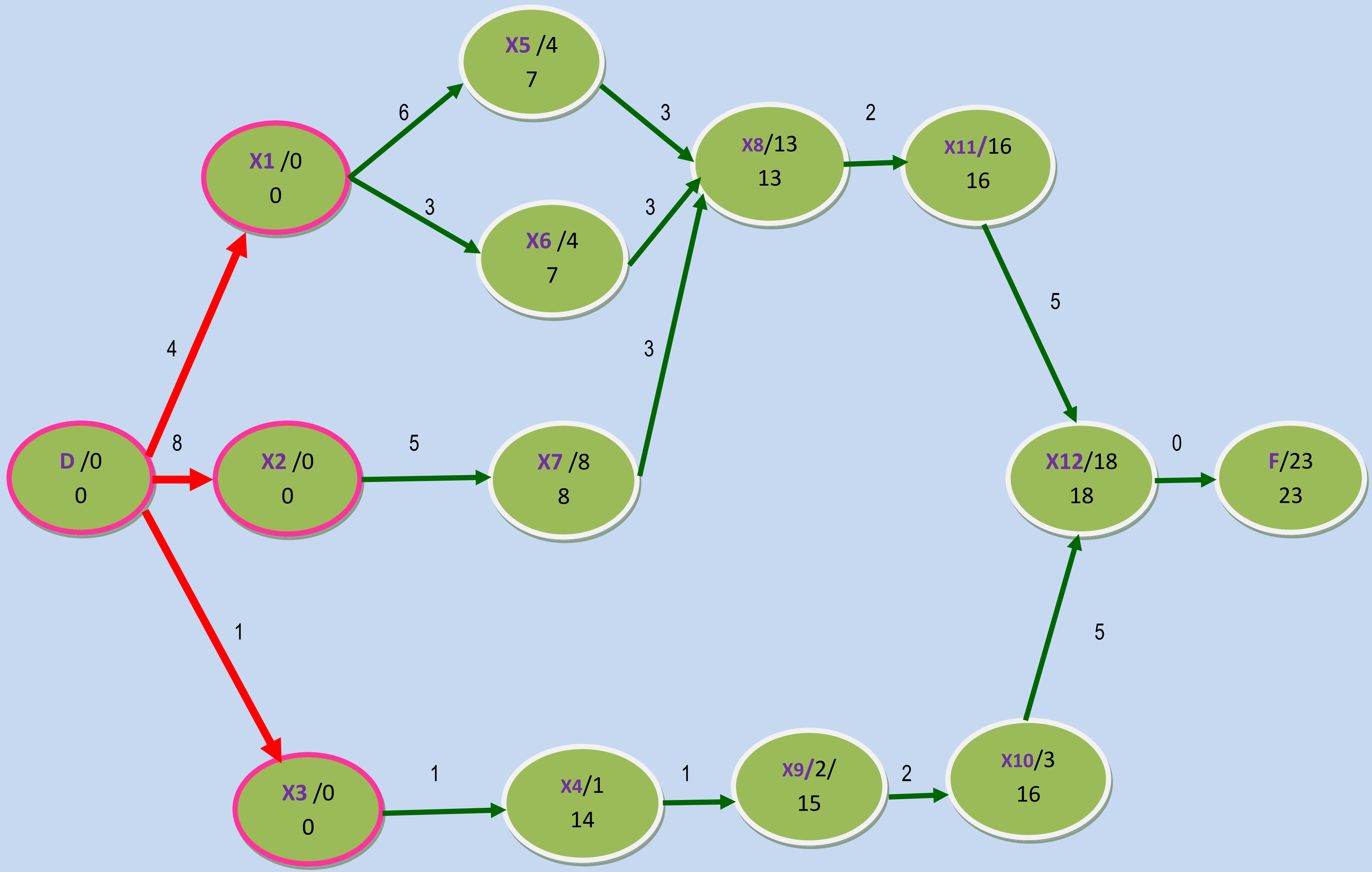
La procédure la plus simple consiste :

- partant du sommet **DEBUT** :

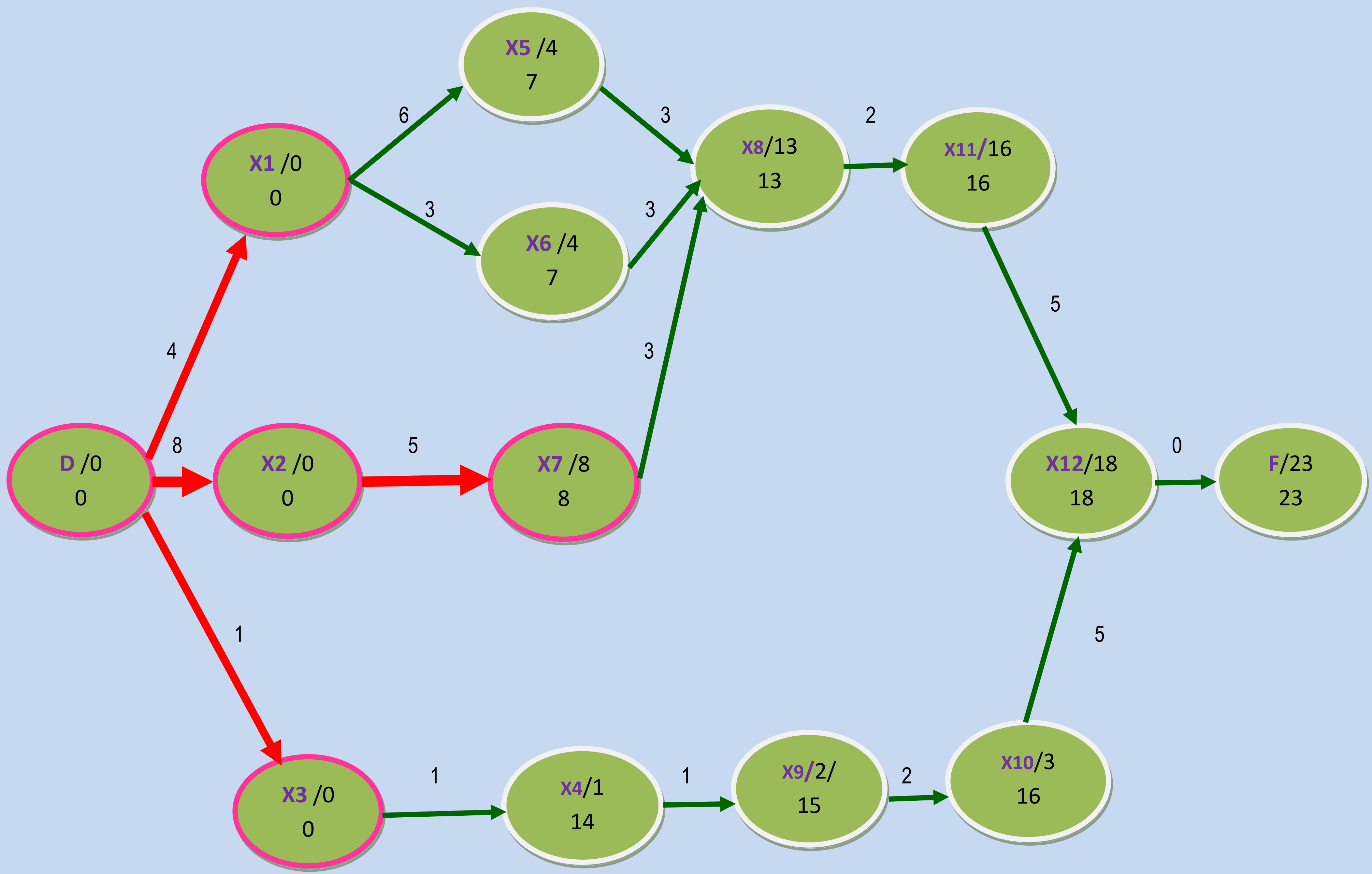
-à **marquer** les sommets s tels que la date au plus tôt est égale à la date au plus tard :

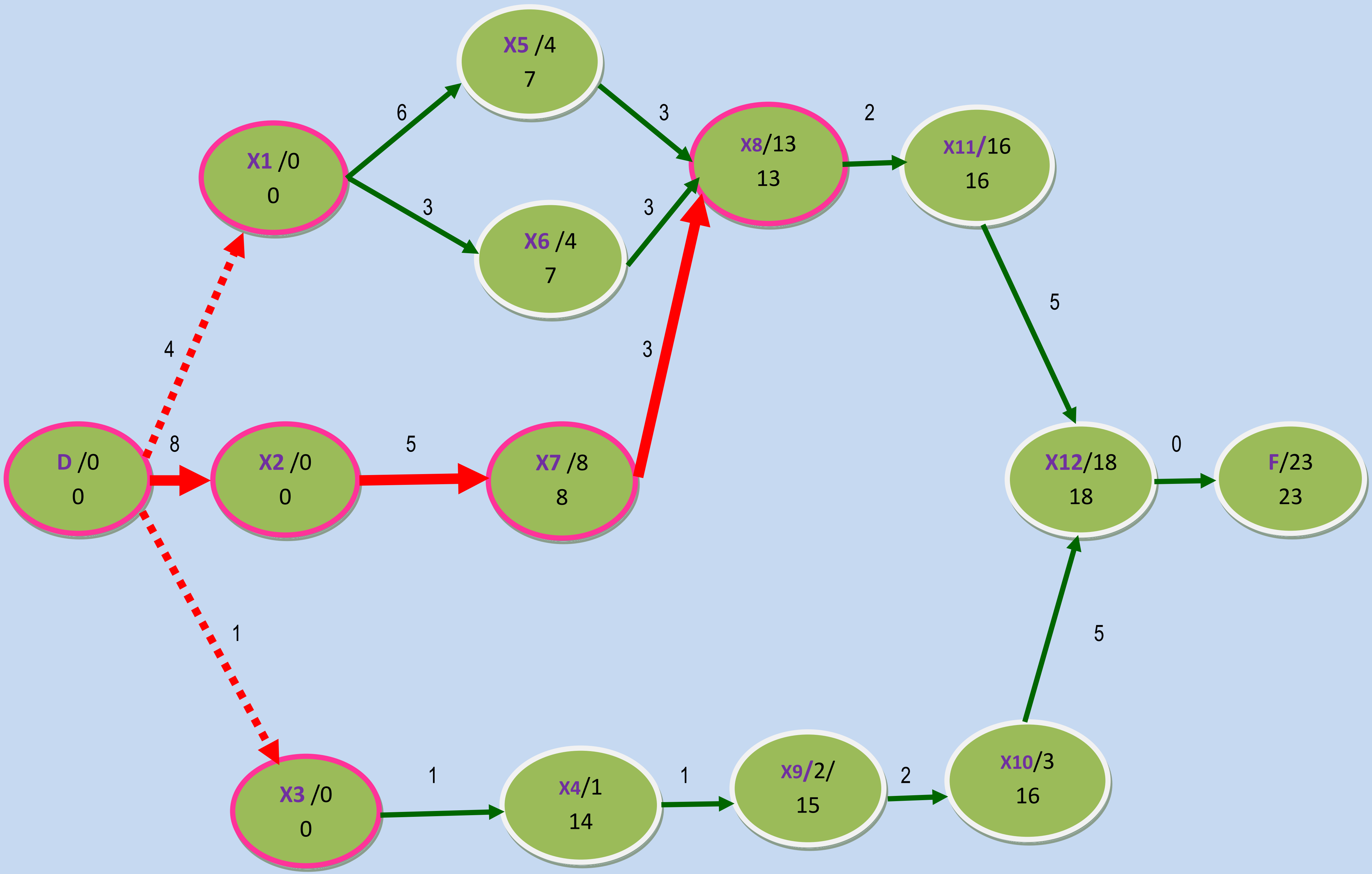
$$Dtôt(s) = Dtard(s)$$

La procédure s'arrête lorsque le sommet **FIN** est **marqué**

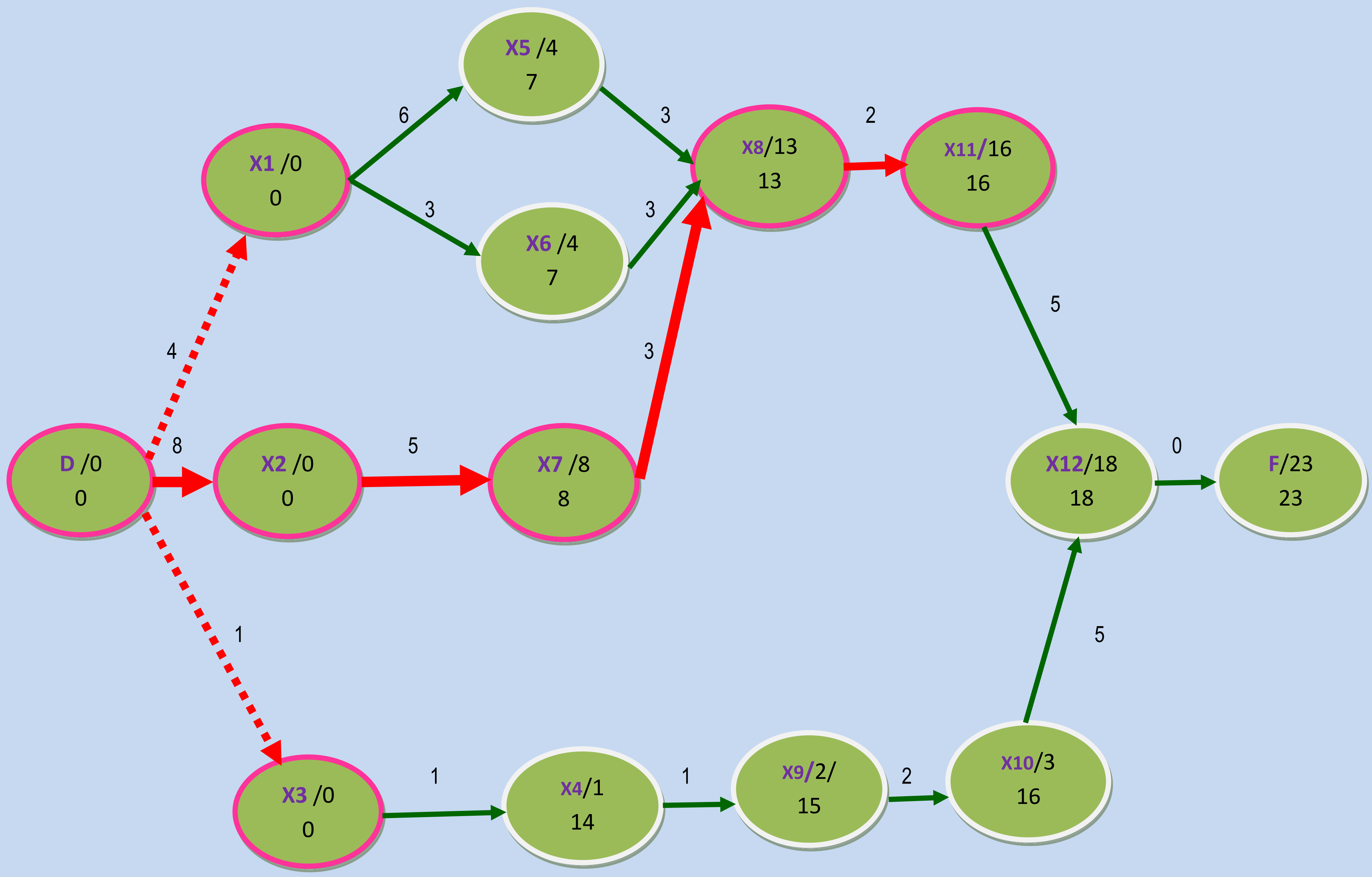


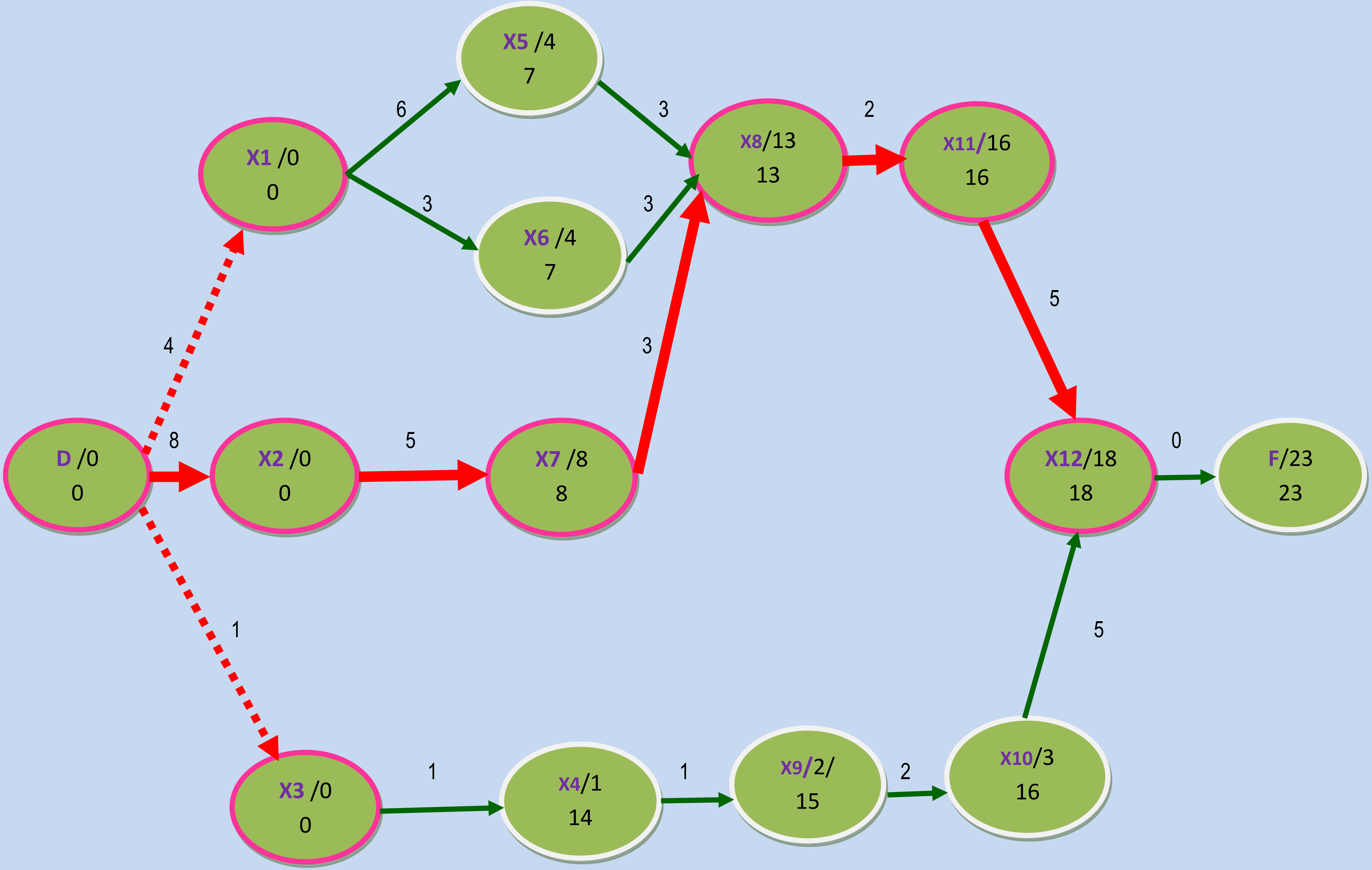
X5 /4
7
X1 /0
0
X6 /4
7
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2/
15
X10/3
16
6
3
3
3
2
4
8
5
3
5
0
1
1
1
2
5

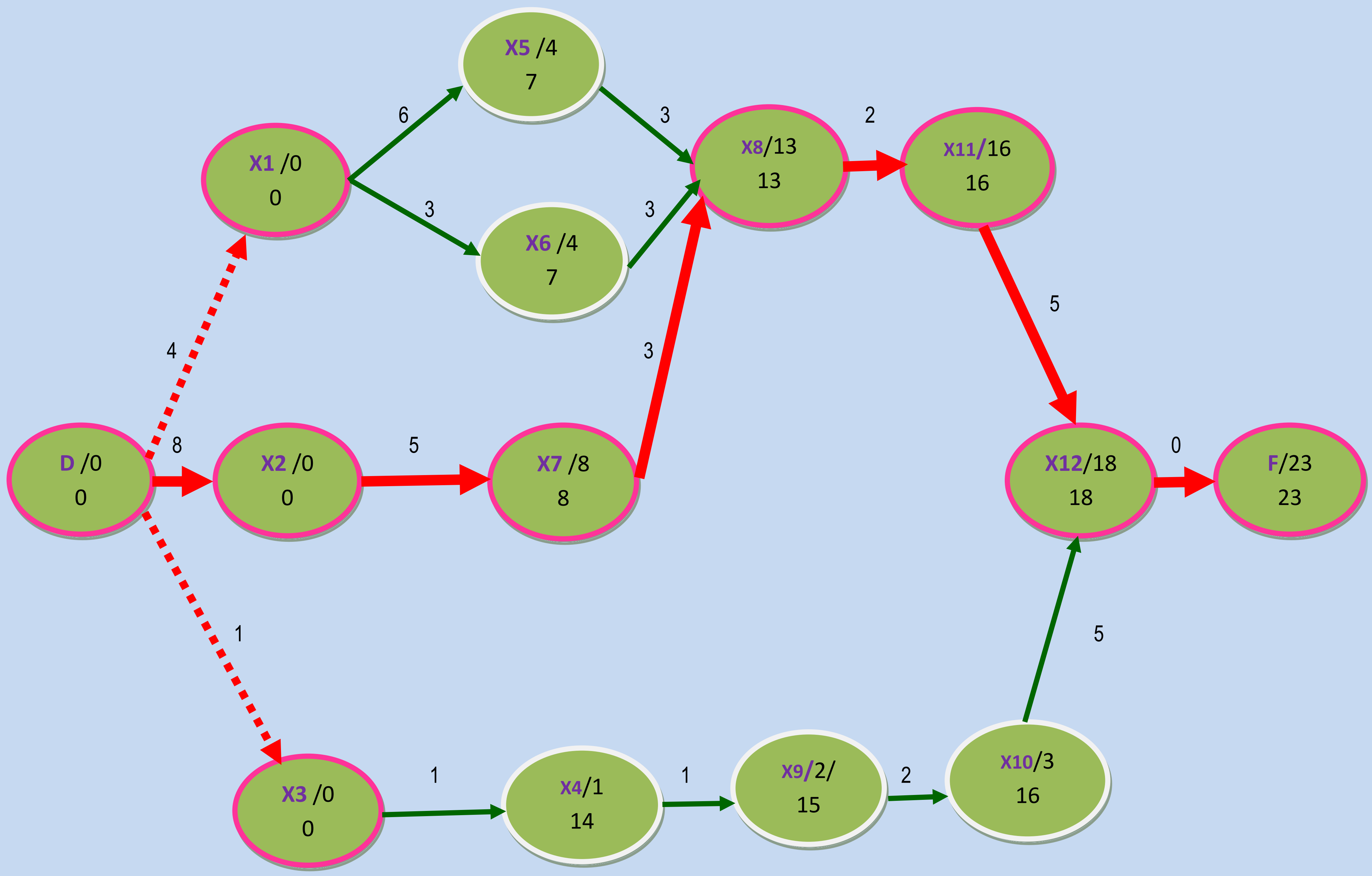

X5 /4
7
X1 /0
0
X6 /4
7
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2/
15
X10/3
16
6
3
3
3
2
4
3
5
8
5
0
1
5
1
1
2



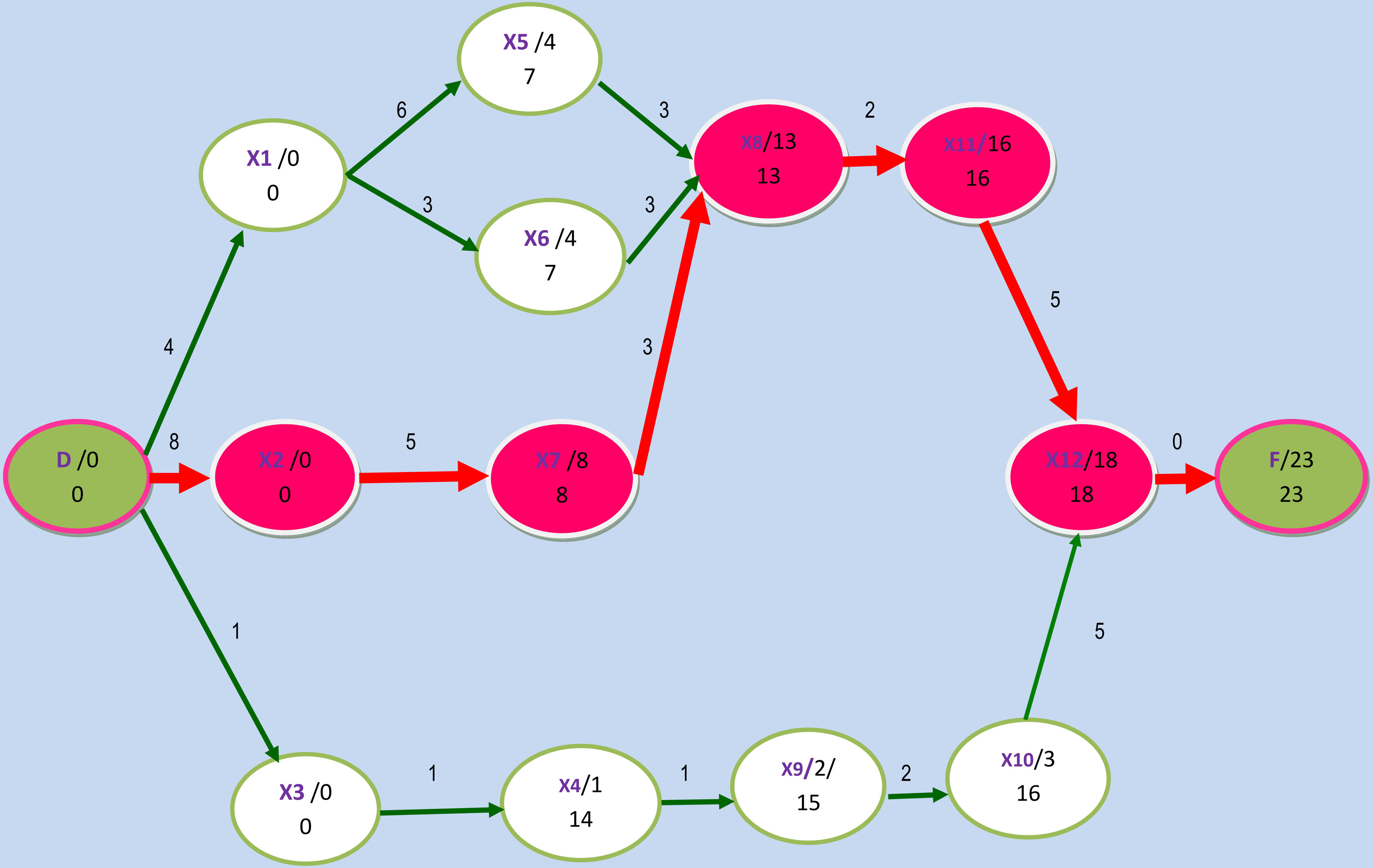
D /0
0
X1 /0
0
X2 /0
0
X3 /0
0
X5 /4
7
X6 /4
7
X7 /8
8
X4/1
14
X8/13
13
X9/2/
15
X10/3
16
X11/16
16
X12/18
18
F/23
23
4
8
1
6
3
5
3
3
3
1
1
2
2
5
5
0


X5 /4
7
X1 /0
0
X6 /4
7
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2/
15
X10/3
16
6
3
3
3
2
4
3
5
8
5
0
1
5
1
1
2


D /0
0
X1 /0
0
X2 /0
0
X3 /0
0
X4/1
14
X5 /4
7
X6 /4
7
X7 /8
8
X8/13
13
X9/2/
15
X10/3
16
X11/16
16
X12/18
18
F/23
23
4
8
1
6
3
5
3
3
3
1
1
2
2
5
5
0



X5 /4
7
X1 /0
0
X6 /4
7
X8/13
13
X11/16
16
D /0
0
X2 /0
0
X7 /8
8
X12/18
18
F/23
23
X3 /0
0
X4/1
14
X9/2/
15
X10/3
16
6
3
3
3
2
4
3
5
8
5
0
1
5
1
1
2



D /0
0
X1 /0
0
X2 /0
0
X3 /0
0
X4/1
14
X5 /4
7
X6 /4
7
X7 /8
8
X8/13
13
X9/2/
15
X10/3
16
X11/16
16
X12/18
18
F/23
23
4
8
1
6
3
5
3
3
3
2
5
1
1
2
5
0

Comparer la durée de parcours de chemin critique et la durée optimale du cycle d'exécution.

Le chemin critique est le **chemin le plus long** entre les sommets DEBUT et FIN.

Il signifie que la tâche FIN est atteinte à partir de la tâche DEBUT au **plus tôt** à l'issue d'une durée **23** qui garantit que n'importe laquelle des tâches est exécutée.

Conclusion1:

Le chemin critique fixe une **durée minimale** pour le cycle d'exécution.

Par ailleurs, il signifie également que la tâche FIN peut être atteinte **au plus tard** à l'issue de la durée calculée le long du chemin le plus long

Conclusion 2:

Le chemin critique fixe une **date au plus tard** pour la fin du cycle d’exécution.

Conclusion 3:

La durée du **chemin critique** détermine la durée **optimale** cycle d’exécution.

Quelle conclusion peut-on alors établir?.

La **durée optimale** du cycle d'exécution est calculée le long du **chemin critique**.