

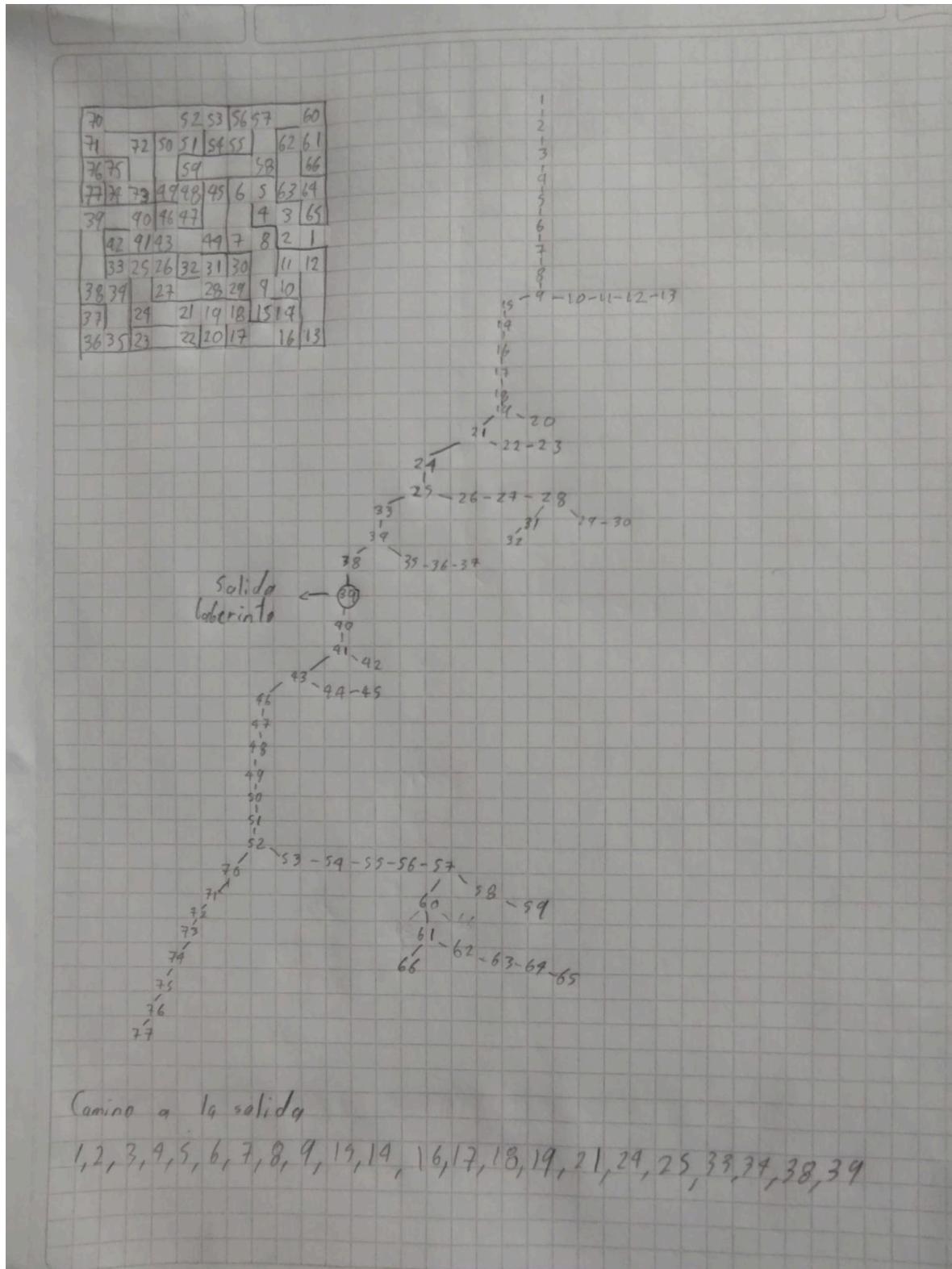


## **Introducción a IA**

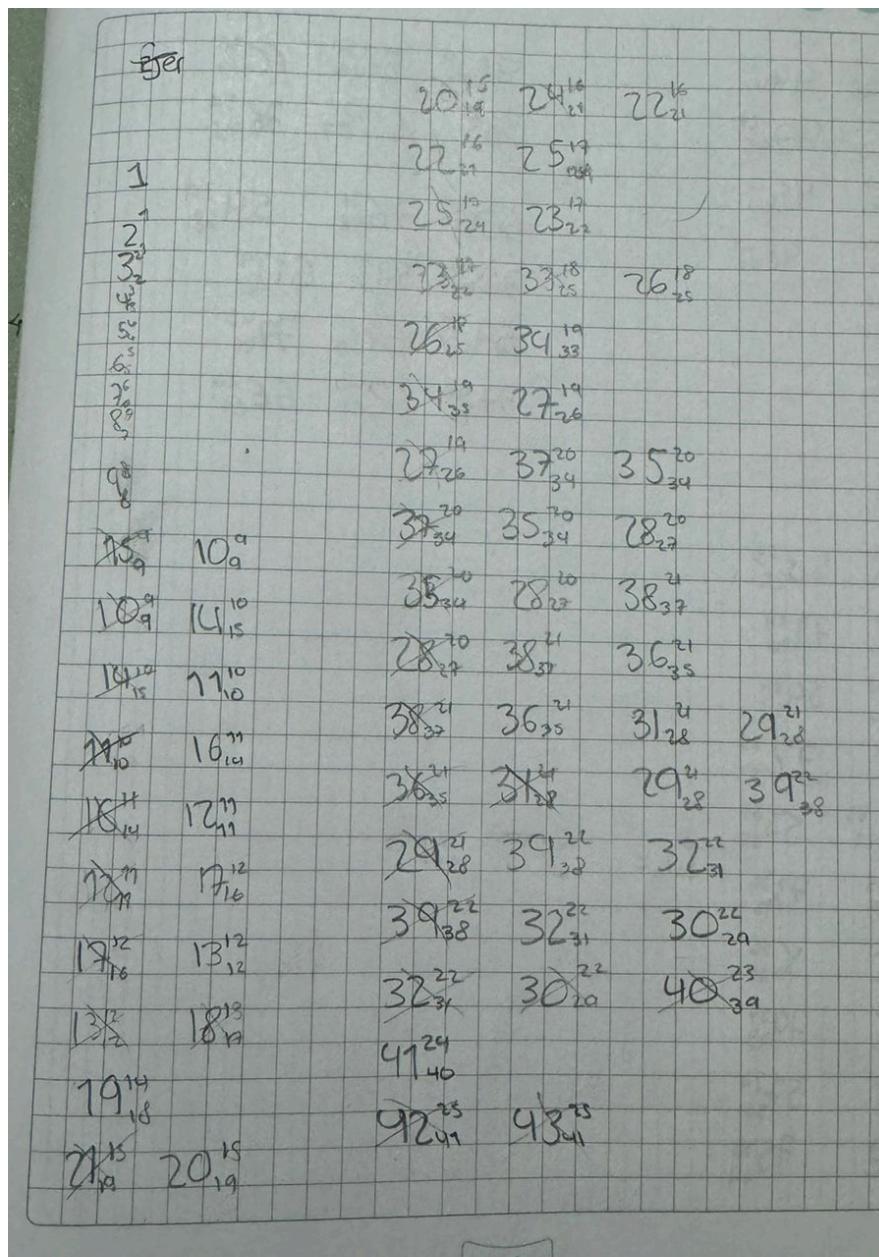
### **Taller 1**

**Samuel Jesus Mejia  
Juan David Barajas Urrea  
Felipe Bolivar Mancilla**

1. Árbol de expansión para el algoritmo en búsqueda en profundidad:



## 2. Árbol de expansión para el algoritmo en búsqueda en anchura.



D	M	A
<del>48</del> <sup>26</sup> <sub>43</sub>	44 <sup>26</sup> <sub>43</sub>	
<del>98</del> <sup>26</sup> <sub>43</sub>	47 <sup>27</sup> <sub>46</sub>	
<del>97</del> <sup>23</sup> <sub>46</sub>	45 <sup>23</sup> <sub>44</sub>	
<del>48</del> <sup>29</sup> <sub>44</sub>	48 <sup>28</sup> <sub>42</sub>	
49 <sup>29</sup> <sub>43</sub>		
50 <sup>30</sup> <sub>49</sub>		
51 <sup>31</sup> <sub>50</sub>		
52 <sup>32</sup> <sub>51</sub>		
<del>30</del> <sup>33</sup> <sub>52</sub>	53 <sup>33</sup> <sub>52</sub>	
<del>58</del> <sup>33</sup> <sub>52</sub>	71 <sup>34</sup> <sub>50</sub>	
<del>37</del> <sup>34</sup> <sub>50</sub>	54 <sup>34</sup> <sub>53</sub>	
<del>51</del> <sup>34</sup> <sub>53</sub>	72 <sup>35</sup> <sub>51</sub>	
<del>38</del> <sup>35</sup> <sub>51</sub>	55 <sup>35</sup> <sub>54</sub>	
<del>55</del> <sup>35</sup> <sub>54</sub>	73 <sup>36</sup> <sub>52</sub>	
<del>38</del> <sup>36</sup> <sub>52</sub>	56 <sup>36</sup> <sub>55</sub>	
<del>56</del> <sup>36</sup> <sub>55</sub>	74 <sup>37</sup> <sub>53</sub>	
<del>34</del> <sup>37</sup> <sub>53</sub>	57 <sup>37</sup> <sub>56</sub>	
<del>57</del> <sup>37</sup> <sub>56</sub>	75 <sup>38</sup> <sub>54</sub>	

### **3. Qué diferencias se encuentran en ambos métodos:**

R: La búsqueda en profundidad (DFS) y la búsqueda en anchura (BFS) se diferencian principalmente en su forma de explorar el árbol de expansión. DFS profundiza en una rama antes de retroceder y explorar otras, utilizando una pila para mantener el rastro de los nodos. Esto puede ser eficiente en términos de memoria, pero no garantiza encontrar la solución más corta y puede quedarse atrapado en ramas profundas. En contraste, BFS explora todos los nodos a un nivel antes de pasar al siguiente, utilizando una cola para gestionar los nodos. Esto asegura encontrar la solución más corta si existe una, pero puede requerir mucha memoria.